

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## GENERÁTOR OTISKŮ PRSTŮ

DIPLOMOVÁ PRÁCE

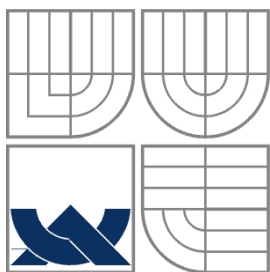
MASTER'S THESIS

AUTOR PRÁCE

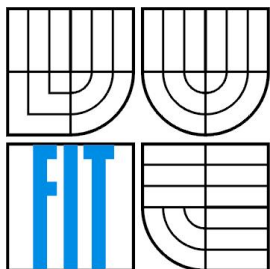
AUTHOR

RADEK CHALOUPKA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# GENERÁTOR OTISKŮ PRSTŮ

FINGERPRINTS GENERATOR

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

RADEK CHALOUPKA

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MARTIN DRAHANSKÝ, Ph.D.

BRNO 2007

## **Zadání**

1. Nastudujte literaturu zabývající se rozpoznáváním otisků prstů. Zaměřte se na metodu používající markanty.
2. Navrhněte metodu generování otisku prstu (průběhu papilárních linií) na základě zadaných markantů.
3. Implementujte navrženou metodu.
4. Dosažené výsledky zhodnoťte.

## **Licenční smlouva**

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

## **Abstrakt**

Algoritmy pro rozpoznávání otisků prstů jsou známé již dlouho a přetrvává snaha o jejich pokud možno co nejlepší optimalizaci. Tato diplomová práce je ovšem na postup opačnou, kdy otisky prstů nejsou rozpoznávány, ale jsou generovány na základě pozic markantů. Takový algoritmus je tedy oprostěn od detekce markantů z obrazu a úprav otisku. Výsledkem práce je aplikace, která generuje otisky prstů na základě několika parametrů, obzvláště markantů.

## **Klíčová slova**

Otisk prstu, markant, papilární linie, závit, smyčka, delta, oblouk, klenutý oblouk, singularita, segmentace, binarizace, zakončení, rozdělení, pole orientací, generování, gaborův filtr

## **Abstract**

Algorithms for fingerprints recognition are already known for long time and there is also an effort for their best optimization. This master's thesis is dealing with an opposite approach, where the fingerprints are not being recognized, but are generated on the minutiae position basis. Such algorithm is then free of the minutiae detection from image and enhancements of fingerprints. Results of this work are the synthetic images generated according to few given parameters, especially minutiae.

## **Keywords**

fingerprint, minutiae, ridge pattern, whorl, loop, delta, arch, tented arch, singularity, segmentation, binarization, termination, bifurcation, directional map, generation, gabor filter

## **Citace**

Radek Chaloupka: Generátor otisků prstů, diplomová práce, Brno, FIT VUT v Brně, 2007

# Generátor otisků prstů

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Martina Dražanského, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Radek Chaloupka  
13. května 2007

## Poděkování

Rád bych poděkoval Ing. Martinu Dražanskému za vstřícný přístup, odborné vedení a pomoc při návrhu struktury diplomové práce.

© Radek Chaloupka, 2007.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	1
1 Úvod.....	3
2 Otisky prstů.....	5
2.1 Papilární linie .....	5
2.2 Charakteristiky otisku prstu.....	6
2.2.1 Druhy markantů.....	6
2.2.2 Markanty a jejich užití k identifikaci .....	7
2.3 Klasifikace otisků prstů.....	8
3 Rozpoznávání otisků prstů.....	10
3.1 Body zájmů .....	10
3.2 Orientace papilární linie a hustota rozložení .....	11
3.2.1 Orientace papilární linie.....	12
3.2.2 Hustota rozložení.....	12
3.3 Segmentace – izolování od pozadí.....	13
3.4 Detekce singularit .....	13
3.5 Vylepšení obrazu a binarizace.....	15
3.6 Detekce markantů .....	16
3.7 Porovnání otisků .....	19
4 Generování otisků prstů.....	21
4.1 Metoda SFinGe.....	21
4.1.1 Generování tvaru otisku.....	21
4.1.2 Generování pole orientací .....	22
4.1.3 Generování rozložení hustoty.....	23
4.1.4 Generování vzorku otisku prstu.....	23
4.1.5 Další případné úpravy .....	24
4.1.6 Příklady výsledků metody SFinGe .....	24
4.2 Vlastní návrh řešení projektu a jeho algoritmus .....	25
4.2.1 Uvedení do problematiky vlastního řešení.....	25
4.2.2 Pole orientací a jeho generování.....	25
4.2.3 Rozložení hustoty .....	28
4.2.4 Možná řešení generování vzoru.....	29
4.2.5 Gaborovy filtry .....	31
4.2.6 Generování papilárních linií.....	33
4.2.7 Selekce zobrazitelné části otisku .....	36

5	Řešení projektu .....	37
5.1	Programovací jazyk C# a jeho možnosti a užití v projektu.....	37
5.2	Možnosti vstupu a výstupu.....	38
5.2.1	Vstup aplikace .....	38
5.2.2	Výstup aplikace .....	39
5.3	Nastavitelné parametry .....	39
5.4	Výsledná aplikace a zhodnocení výsledků .....	40
5.5	Návrhy pro další možnosti aplikace a její další vývoj .....	41
6	Závěr .....	43
	Literatura.....	45
	Seznam příloh .....	46
	Seznam obrázků .....	47

# 1 Úvod

Tato diplomová práce se zabývá problematikou otisků prstů a následně možností otisky generovat na základě získaných teoretických znalostí o otiscích prstů a metodách, se kterými pracuje softwarové rozpoznávání otisků. Toto generování se bude provádět na základě předem zadaných markantů, které v sobě nesou informaci o své poloze, typu a úhlu natočení v otisku, kde úhel reprezentuje směr, kterým se ubírá papilární linie v daném místě.

Oficiálně existuje jedna metoda pro generování otisků prstů z italské univerzity v Boloni, která však negeneruje otisky na základě markantů, ale podle nadefinování singularit a tvaru otisku, což je mnohem jednodušší, než vytváření otisků na základě markantů. Tato metoda už se ovšem nějaký čas vyvíjí a v dnešní době se již pracuje na třetí verzi. Aplikace z této univerzity vrací uspokojivé a celkem věrné napodobeniny otisků prstů, které se velmi blíží reálnému otisku zanechanému nějakým člověkem. Je to způsobeno tím, že na vygenerovaný otisk ještě aplikují filtry zajišťující věrnější vzhled. Poslední dostupná verze však není příliš uživatelsky přívětivá a vytvoření otisku díky ne moc propracovanému graficky uživatelskému rozhraní není nejšťastnější.

Výsledkem práce tedy bude aplikace s pokud možno co nejpřehlednějším graficky uživatelským prostředím a dále návrh a realizace algoritmu, který v této aplikaci bude implementován a který bude umožňovat generování otisků prstů podle předem zadaných markantů. Vytvořený otisk by měl reprezentovat průchod papilárních linií bez nutnosti dalších postprocesorových úprav otisku pro zajištění jeho věrnější ztvárnění.

První kapitola se věnuje otiskům a jejich základním vlastnostem, aby se podle těchto charakteristik dalo dále pracovat a algoritmicky je zpracovávat.

Druhá kapitola je zaměřena způsoby rozpoznávání otisků prstů za pomoci počítačové techniky. V jednotlivých podkapitolách se postupuje přibližnými kroky představující postup algoritmu a zahrnuje operace, které se s otisky provádí až do finálního porovnání otisků. Tato kapitola je pro seznámení s problematikou také důležitá, protože některé rozpoznávací techniky a postupy umožní lepší pochopení a lepší návrh postupu, kdy se při generování otisků bude postupovat obdobně, jen v opačném směru.

Třetí kapitola je zaměřena na samotné generování otisků prstů, kdy je v první podkapitole popsána metoda SFinGe z italské univerzity a jejich aplikace. Druhá podkapitola obsahuje možné návrhy vlastních algoritmů, výhody a nevýhody těchto metod, problematičnost realizace a nakonec vybrání jednoho postupu pro generování otisku prstu podle předem daných markantů a rozepsání tohoto algoritmu.

Ve čtvrté kapitole je popsán způsob implementace aplikace a výhody programovacího jazyka, ve kterém byla aplikace vytvořena. Dále jsou ve stručnosti shrnuty možnosti a vlastnosti aplikace.



Kapitola je zakončena ukázkou aplikace, možnými výstupy a také případnými návrhy na další rozvoj a zdokonalování aplikace.

Pátá a poslední kapitola pojednává o biometrických systémech a jejich rozšiřování ve světě. Dále o vývoji v rozpoznávání otisků prstů v kombinaci s výsledky, které byly dosaženy v této práci. Kapitola je zakončena popisem směru, kterým se ubírá toto odvětví biometrických systémů a další uplatnění v budoucnosti.

## 2 Otisky prstů

Již v dávné minulosti si lidé všimli, že jejich kůže se vyznačuje jistými vlastnostmi a charakteristikami a že každý člověk má na prstech specifické otisky.

Postupem času bylo zjištěno, že tyto otisky se člověk od člověka liší svým vzhledem a že je dokonce mizivá pravděpodobnost, že by existovali dva lidé se dvěma stejnými otisky prstů. Díky této vlastnosti se problematice otisků prstů začalo věnovat stále více a více lidí z důvodu identifikace osob a to nejprve v oboru kriminalistiky, kdy je při šetření případů trestné činnosti jednou z nejdůležitějších věcí správná identifikace a korektní rozpoznání pachatele jako konkrétního člověka a původce zločinu.

### 2.1 Papilární linie

Na vnitřní straně rukou a chodidel každého člověka jsou již od narození vidět útvary vytvořené papilárními liniemi.

Vytvářejí se již ve čtvrtém měsíci vývoje dítěte a do konce života zůstávají neměnné [1]. Tyto papilární linie vytvářejí složité a svým způsobem jedinečné obrazce, jejichž přesný význam a způsob vytváření se nepodařilo doposud plně objasnit. Je ovšem jasné, že mají co do činění s citlivostí na dotek a také díky nim dochází k lepším hmatovým vlastnostem při styku s objekty, se kterými se manipuluje. Papilárními liniemi nejsou obdařeni jenom lidé, ale vyskytují se i u opic. Bylo zjištěno, že u některých opic, které mají chápavý ocas, se papilární linie vyskytují i na dolní části jejich ocasu.

Reliéfy, které papilární linie vytváří, dosahují na končetinách výšky v rozmezí 0,1 – 0,4 mm a šířky 0,2 – 0,7 mm.

V průběhu lidského života jsou tyto linie relativně neměnné. Tato relativní neměnnost spočívá v tom, že během růstu člověka sice dochází k velikostním změnám otisků, popřípadě k výskytu vrásek, avšak sled a návaznost papilárních linií zůstává zachována.

Další vlastností papilárních linií je, že během života člověka nejdou odstranit, tedy pokud nedojde k přímému odstranění zárodečné vrstvy kůže. Při běžných mechanických poškozeních se papilární linie vždy postupem času obnoví do původního vzhledu. Pokud dojde ke spálení, chemickému popálení, sedření nebo seříznutí povrchové vrstvy kůže, papilární linie zůstanou zachovány a postupem času opět dochází k regeneraci. Vše tedy závisí na případném poškození zárodečné vrstvy kůže, která stojí za uchováním předem daných hodnot. Pokud k poškození dojde, vznikají v otisku prstu trvalé jizvy, čímž dojde k dalšímu charakteristickému vzhledu.

## 2.2 Charakteristiky otisku prstu

Díky zkoumání a pozorování otisků bylo vysledováno několik charakteristických znaků, ze kterých se každý otisk skládá. Těmto znakům se říká markanty a představují klíčovou informaci, která se používá pro identifikaci.

Budou se rovněž používat jako klíčová informace pro generování otisku prstu, protože svým způsobem kompletně popisují sled a uspořádání papilárních linií na otisku a tím pádem v případě, kdy bychom měli informace o všech markantech z otisku, měli bychom být schopni rekonstruovat otisk bez větších problémů. Všechny markanty z otisku však k dispozici prakticky nikdy mít nebudeme, proto se musíme spokojit s takovým počtem markantů, kolik budeme mít na vstupu.

### 2.2.1 Druhy markantů

Jednotlivé charakteristické znaky, neboli markanty se rozdělují do několika druhů [1].



**Obrázek 2.1: Základní typy markantů**

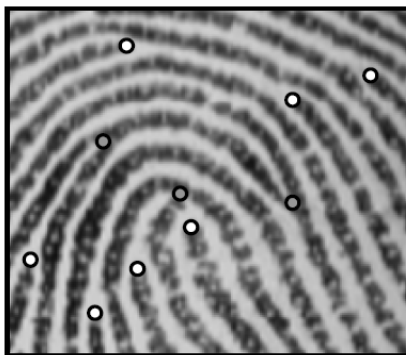
Během výzkumů bylo zjištěno, že každý druh markantu má nějakou pravděpodobnost, se kterou se vyskytuje ve zkoumaném otisku. Kromě již zmíněných typických druhů markantů lze rozlišit svým charakteristickým tvarem i další druhy, jako je například přerušení, ukončení ve vidlici a podobně. Po zamyšlení ale dojdeme k závěru, že všechny markanty jsou vlastně podmnožinou dvou hlavních a to markantu začátku (ukončení), představující počátek, popřípadě konec papilární linie a markantu vidlice, představující rozvětvení papilární linie. Díky těmto základním markantům se dají popsat i markanty další.

## 2.2.2 Markanty a jejich užití k identifikaci

Použití jen dvou základních typů markantů pro popis jednotlivých otisků má hlavní důvod ve zjednodušení procesu rozpoznávání pomocí počítače, protože jejich detekce je z obrázku jednoznačná a triviální.

V případě, že bychom se pokoušeli detekovat nějaký složitější markant, ovlivnilo by to výpočetní náročnost tohoto úkonu, což je pro nás velmi nežádoucí hlavně v případě, když hledáme identický otisk ve skupině několika dalších otisků.

Pro identifikaci se používají souřadnice těchto významných markantů a tím jsme schopni porovnáním dvou otisků zjistit, zda se jedná o otisky shodné či nikoli. Svým způsobem se tedy ve finále jedná o číselnou hodnotu, která daného člověka identifikuje mezi všemi ostatními lidmi. K identifikaci osob se například používají i rodná čísla. Ta ovšem v porovnání s čísly vygenerovanými z otisku prstu nemusí být jednoznačná. Čísla generovaná z otisku prstu jsou použitelná v globálním měřítku, protože podstata a charakteristika otisků prstů je shodná pro všechny lidi na Zemi.



**Obrázek 2.2:** Část otisku s detekovanými markanty

Identifikace člověka pomocí otisku prstů spočívá v nalezení určitého množství shodných markantů. Toto množství se stát od státu liší v závislosti na dříve prosazených pravidlech pro identifikaci. V některých státech se již ani nepožaduje fixní počet shodných markantů, ale přihlíží se k více okolnostem.

Shody se vyhledávají algoritmicky, kdy po nalezení možné shody dochází většinou ještě k porovnání otisků člověkem. To se týká identifikace osob v kriminalistice, kdy je v databázi uložen navíc i celý pořízený otisk prstu. Zahrnutím lidského faktoru do tohoto procesu ovšem vede k riziku, kdy člověk finální kontrolu zanedbá a díky nekompletnímu otisku prstu může odsouhlasit shodu nesprávně, což dokládá již několik příkladů z minulosti v různých justičních omylech. V těchto případech však selhal člověk, ne metoda jako taková.

## 2.3 Klasifikace otisků prstů

Identifikace osoby představuje porovnání otisku prstu se všemi otisky v databázi, což by při rozsáhlých databázích představovalo obrovské množství otisků. Ve finále by to znamenalo velkou výpočetní náročnost pro nalezení nebo vyloučení shody v databázi.

K redukování počtu porovnání a ke zrychlení této operace jsou otisky rozděleny do předem definovaných tříd. Klasifikace otisků prstů představuje zařazení každého otisku do odpovídající třídy, takže hledaný otisk bude porovnáván pouze s podmnožinou z celé databáze, která odpovídá příslušné třídě. I přesto je však prohledávaných dat velké množství a toto prohledávání je časově náročné, i když pro svoji úspěšnost postačující.

Všechny klasifikační schémata, která v dnešních dobách využívá policie po celém světě, jsou odvozena z takzvané Henryho klasifikace [1]. Tato klasifikace obsahuje 3 základní typy otisků:

- arch (oblouk)
- loop (smyčka)
- whorl (závit)

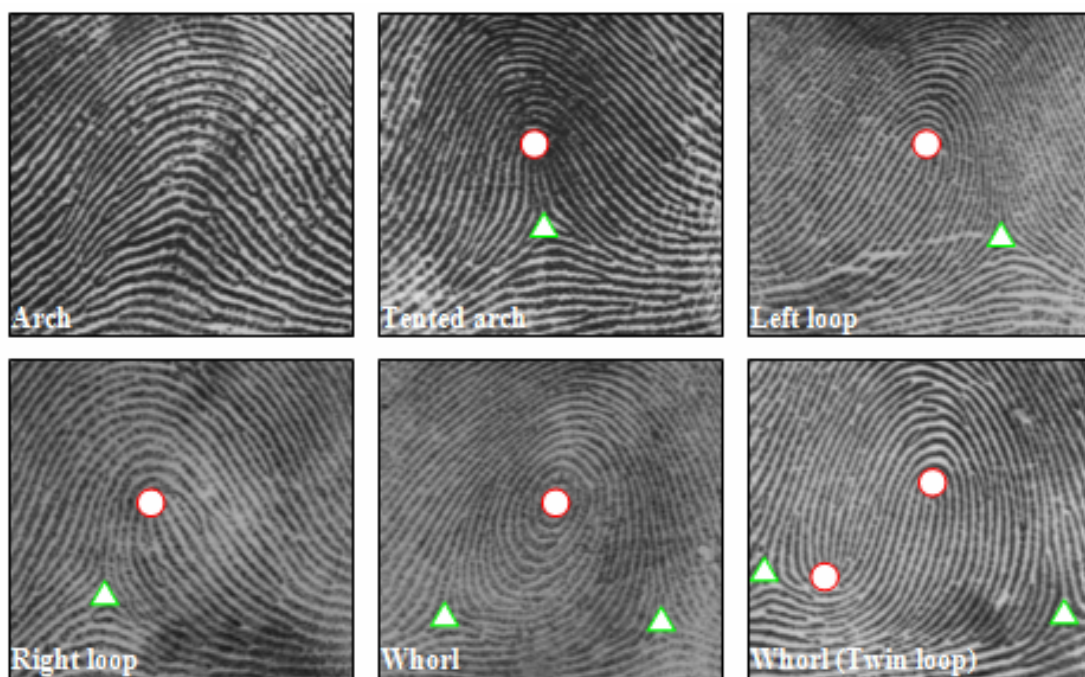
V dnešní době se ale většinou používá klasifikace s 5-ti třídami (arch, tented arch, left loop, right loop, whorl). Ve skutečnosti však otisky prstů v těchto třídách nemají rovnoměrné zastoupení. Po přepočtu náhodného vzorku bylo zjištěno rozložení takovéto [6]:

- arch (oblouk)	3,7%
- tented arch (klenutý oblouk)	2,9%
- left loop (levá smyčka)	33,8%
- right loop (pravá smyčka)	31,7%
- whorl (závit)	27,9%

Henryho klasifikace byla postupně přepracována na mnohem podrobnější a v dnešní době je známa pod zkratkou NCIC. Otisky uložené v databázi jsou označeny čísly od 1 do 10, kdy číslo 1 představuje levý palec a číslo 10 pravý malíček. Každý otisk je dále kódován dvěma znaky a tento dvouznačkový kód obsahuje informaci o třídě otisku a další dodatečné informace jako je např. počet papilárních linií mezi body jádra a bodem delty. Těchto deset dvojnákových kódů tvoří dvacetiznačkové pole, kterému se říká FPC (Fingerprint Classification file), které slouží jako klíč k prohledávání databáze otisků prstů [10].

Na obrázku 2.3 je znázorněno 6 základních typů otisků podle pozice singularit. První typ oblouk se vyznačuje tím, že neobsahuje žádnou singularitu, pouze specifické zakřivení dle tvaru prstu. Klenutý oblouk je velmi podobný prvnímu typu, avšak už obsahuje singularitu jádra a

singularitu delty. Singularity jsou ve vertikální poloze, proto je výsledný tvar podobný prvnímu typu. U ostatních typů už je delta oproti jádru posunutá buď vlevo nebo vpravo a z těchto kombinací jsou vytvořeny další dva typy a to levá a pravá smyčka. Závit už obsahuje delty dvě, případně může obsahovat i dvě jádra.



**Obrázek 2.3: Základní typy otisků prstů podle rozložení singularit**

## 3 Rozpoznávání otisků prstů

Pokud chceme dobře porozumět metodám a postupům generování otisků prstů, je rozumné si nejdříve osvojit techniky, které se používají při jejich rozpoznávání. Svým způsobem se totiž jedná o vzájemně opačné postupy.

Při rozpoznávání otisků buď porovnáváme otisk s jiným otiskem nebo daný otisk hledáme v nějaké databázi, což je již časově daleko náročnější.

Abychom mohli otisk zpracovat, musíme ho nejdřív nějak získat. Na to existuje několik metod v závislosti na možnostech a přístupu, kdy můžeme prst přitisknout na snímač a získat ho tak ihned z prstu člověka a nebo pomocí různých technik dostat již z nějakého předmětu, kde byl otisk zanechán a následně naskenovat pro další počítačové zpracování. Pokud již máme otisk k dispozici, můžeme přejít k dalším krokům.

Tyto níže popsané kroky jsou stručným a zjednodušeným postupem a nejedná se o jediný a správný postup, který lze použít, ale zato je to postup nejvíce rozšířený, který poslouží k samotnému pochopení generování nejlépe. Spousta metod lze nějakým způsobem modifikovat nebo některé kroky upravit. Dále se různé postupy mohou lišit v závislosti na zemi, ve které se otisk rozpoznává, i když vlivem globalizace je snaha o ucelení a sjednocení metod při rozpoznávání a zdokonalení těchto metod takovým způsobem, aby docházelo k rozpoznávání s co nejlepší přesností a pokud možno v co nejkratším čase.

### 3.1 Body zájmů

Otisky se skládají z charakteristických tvarů, které se dají popsat a mají specifické vlastnosti. Tím pádem se tyto tvary dají také rozeznat a zjistit jejich pozice.

Otisk prstu se skládá ze světlých a tmavých míst, kdy tmavá místa představují otisklé papilární linie z prstu, které jsou vystouplejší a světlá místa představují mezery mezi nimi. Papilární linie se na otisku můžou zakončovat nebo rozdvojovat.

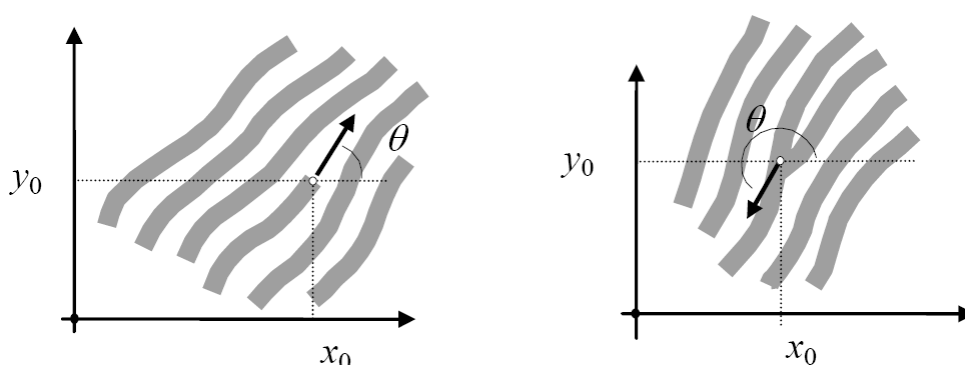
Na některých místech otisku jsou vidět výraznější tvary (charakteristické například zvýšeným výskytem zakřivených papilárních linií nebo větším počtu zakončení). Tyto oblasti (singularity) mohou být klasifikovány do tří typů: loop (smyčka), delta a whorl (závit). Smyčka a závit představují jádro otisku prstu.

Z rozmístění těchto singularit na otisku prstu vzniká několik již zmíněných typů otisků prstů a tudíž jsme po zjištění poloh těchto singularit schopni daný otisk zařadit do patřičného typu otisku.

Dále nás na otisku prstu zajímají markanty. Na různých místech se papilární linie může znenadání ukončit. Toto místo je markant představující zakončení. V dalším případě se může papilární linie rozdvojit, což představuje markant vidlice nebo také rozdvojení. Rozeznat lze i další

typy markantů, ovšem to je pro detekci trochu náročnější, protože patří do podmnožiny markantů zakončení a vidlice.

Dle obecného modelu rozpoznávání otisků ve světě se počítá pouze s markanty zakončení a vidlice, kdy se zjišťuje poloha tohoto markantu a úhel, pod kterým je znázorněn. Určování těchto úhlů je znázorněno na obrázku 3.1, kde je znázorněn směrový vektor papilární linie a úhel, ve kterém tato papilární linie postupuje. V praxi se ještě můžeme setkat z rozdílným definováním úhlu markantu a to takovým, že směrový vektor je otočen o  $180^\circ$ . V této práci a výsledné aplikaci však bude použit tento způsob určení úhlu.



**Obrázek 3.1: Poloha a úhel u markantu typu zakončení a typu vidlice**

Ačkoliv některé metody porovnávání otisků prstů porovnávají otisky přímo na základě korelačních metod, tak je známo, že obrázky ve stupních šedi pro porovnávání nejsou příliš kvalitní reprezentací otisku a může docházet k častějším chybám a nepřesnostem. I u těchto metod však díky počítačové technice dochází k jejich zdokonalování.

Před porovnáním otisku se většinou pro zlepšení rozpoznávacích technik provede několik operací, které původní otisk upraví na otisk lépe zpracovatelný.

## 3.2 Orientace papilární linie a hustota rozložení

Orientace papilární linie představuje úhel, v jakém je natočená na příslušné pozici obrázku otisku prstu a hustota zase definuje vzdálenosti mezi jednotlivými papilárními liniemi.

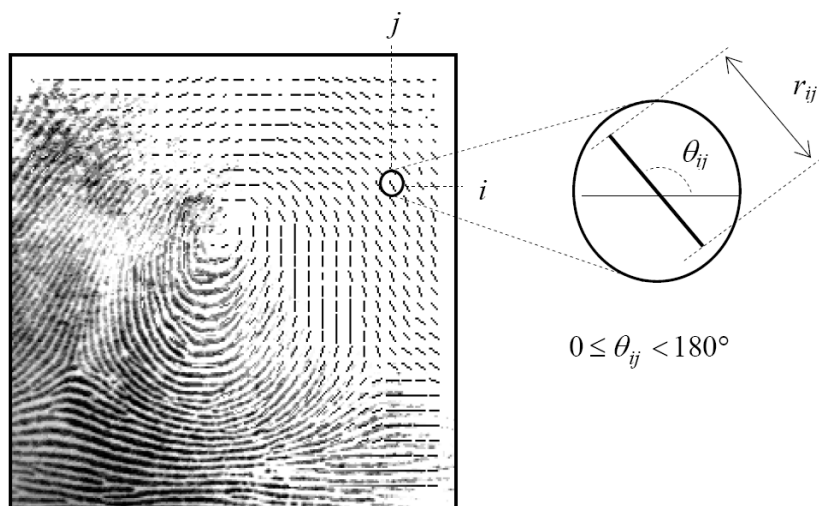
Popsáním těchto charakteristik otisku prstu získáme přibližnou představu o tvaru a rozložení papilárních linií, proto je to také jedna z nejdůležitějších částí pro pozdější algoritmus generování vlastního otisku prstu.

V algoritmech pro rozpoznávání otisků prstů však tyto charakteristiky hrají také nemalou roli a napomáhají při úpravách obrázku otisku pro jeho kvalitnější rozpoznání.



### 3.2.1 Orientace papilární linie

Na obrázku 3.2 vidíme, jak je otisk rozdělen do matice, kdy každý prvek matice znázorňuje orientaci papilární linie na odpovídající pozici. Po aplikování na celý obrázek získáme pole orientací otisku prstu. Můžeme si také všimnout, že v orientaci se nerozlišuje směr papilární linie, ve kterém se rozšiřuje, proto například hodnota orientace  $45^\circ$  a  $225^\circ$  vypadá identicky. Hodnoty orientace tedy nabývají hodnot od  $0^\circ$  do  $180^\circ$ .



Obrázek 3.2: Vytváření pole orientací z papilárních linií

### 3.2.2 Hustota rozložení

Hustota rozložení papilárních linií se počítá také na matici, která rozdělí původní obrázek otisku, a každý prvek matice představuje hustotu papilárních linií na patřičné pozici v obrázku. Hustota rozložení se otisk od otisku liší, i když se v tomto případě dají vypořádkovat jisté podobnosti, kdy například špičky prstu mívají tuto hustotu menší.



Obrázek 3.3: Hustota rozložení u dvou různých otisků prstů

Na obrázku 3.3 vidíme hustotu rozložení, kde světlejší místa představují větší hustotu a tmavá místa naopak hustotu menší. Na obrázcích je pozorovatelná dříve zmíněná menší hustota rozložení u špičky prstu.

### 3.3 Segmentace – izolování od pozadí

Oddělení otisku prstu od pozadí napomáhá při detekci tomu, aby otisk nebyl ovlivňován nečistotami a případnými šumy v pozadí.

Jelikož se otisk prstu skládá z linek představující papilární linie, nebylo by nejvhodnější užití globálních ani lokálních prahových metod. Ve skutečnosti obrázek otisku prstů totiž nejvíce narušují místa, kdy v popředí otisku jsou orientované papilární linie a na pozadí pod otiskem jsou neorientované a náhodné nepřesnosti, které nemají danu žádnou hlavní orientaci.

Pokud je pozadí obrázku vždy světlejší oproti oblasti představující otisk prstu, lze nežádoucí pozadí jednoduše odfiltrovat od otisku použitím filtru na základě intenzity barvy. V případě přítomnosti nečistot a dalších rušivých elementů je však potřeba použít robustnějších metod pro izolování otisku prstu od pozadí.

Toto se však nejvíce týká otisků pořízených na nějakých předmětech, popřípadě při nečistém snímání.

### 3.4 Detekce singularit

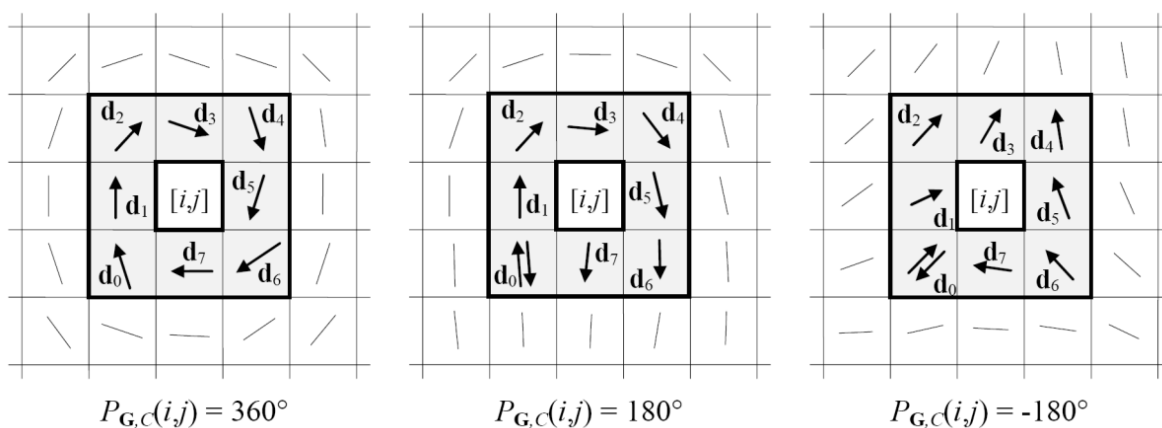
Singularity jsou stěžejní charakteristiky otisků, které nám umožňují otisky prstů rozdělit a zařadit do patřičných typů dle klasifikace těchto singularit.

Pro detekci singularit existuje více metod, nejznámější však je metoda založená na výpočtu Poincaréova indexu [5]. Necht'  $C$  je uzavřená cesta definovaná jako postupná sekvence elementů z pole orientací na souřadnicích  $[i, j]$ , potom Poincaréův index  $P_{G,C}(i, j)$  na  $[i, j]$  se počítá jako algebraický součet rozdílů úhlů sousedních elementů z  $C$ . Ke sčítání rozdílů orientací potřebujeme přiřadit každé orientaci původní směr (máme dvě možnosti). Řešením tohoto problému je náhodný výběr směru prvního prvku a každému následujícímu prvku přiřazení směru, který je svému předchozímu prvku nejbližší.

Je známo a také lze lehce ukázat, že Poincaréův index může nabývat pouze diskrétních hodnot v rozmezí:  $0^\circ$ ,  $\pm 180^\circ$  a  $\pm 360^\circ$ . V případě singularit u otisků prstů může být  $P_{G,C}(i, j)$  rovno:

- $0^\circ$                       pokud  $[i, j]$  neodpovídá žádné singularitě
- $360^\circ$                     pokud  $[i, j]$  odpovídá singularitě typu whorl (závit)
- $180^\circ$                     pokud  $[i, j]$  odpovídá singularitě typu loop (smyčka)
- $-180^\circ$                    pokud  $[i, j]$  odpovídá singularitě typu delta

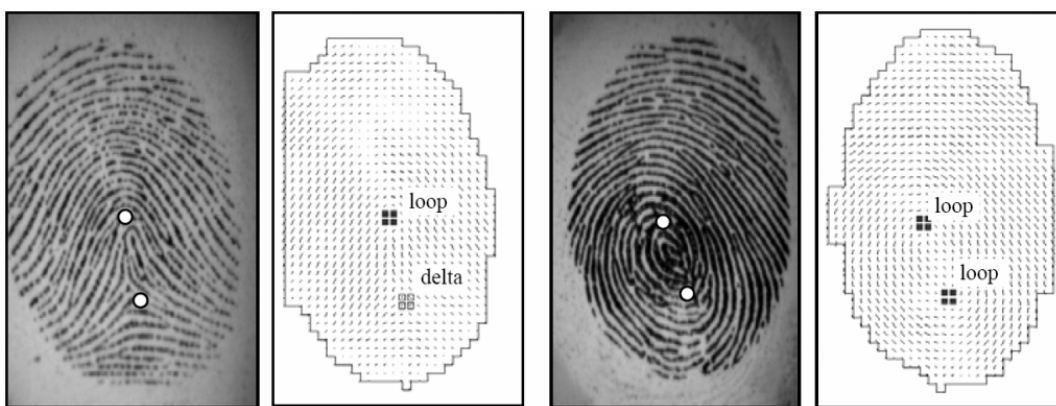
Na obrázku 3.4 vidíme ukázky výpočtu Poincaréova indexu pro singularity závit (whorl), smyčku (loop) a deltu. Zobrazeny jsou 3 části z pole orientací. Cesta definující C je postupná sekvence osmi elementů  $d_k$  ( $k = 0..7$ ) kolem prvku  $[i, j]$ . Směry všech elementů jsou určeny následovně: prvek  $d_0$  směřuje nahoru; směry pro  $d_k$  ( $k = 1..7$ ) jsou orientovány tak, aby absolutní hodnota mezi úhly  $d_k$  a  $d_{k-1}$  byla menší nebo rovna  $90^\circ$ .



**Obrázek 3.4: Detekce singularit pomocí Poincaréova indexu**

Poincaréův index je vypočítán následovně:

$$P_{G,C}(i, j) = \sum_{k=0}^7 \text{úhel}(d_k, d_{(k+1) \bmod 8}) \quad (3.1)$$



**Obrázek 3.5: Příklad detekce singularit na otisku prstu**

Na obrázku 3.5 vidíme dva příklady detekce singularit touto metodou.

Detekce singularit pomocí Poincaréova indexu v obrázku otisku prstu nízké kvality a s výskytem nečistot a nepřesností je většinou nepřesná a může dojít k detekování chybných singularit, které ve skutečnosti na patřičném otisku být nemají. To je zapříčiněno ztíženou detekcí orientace na

příslušném místě v obrázku kvůli nepřesnostem v otisku, kdy není patrný směr papilární linie na daném místě.

Detekce singularit je možné provádět i jinými metodami, ty mohou být založeny na poli orientací, jako tomu bylo u Poincaréova indexu, na metodách zpracujících obraz po částech a na detekci jádra z otisku.

### 3.5 Vylepšení obrazu a binarizace

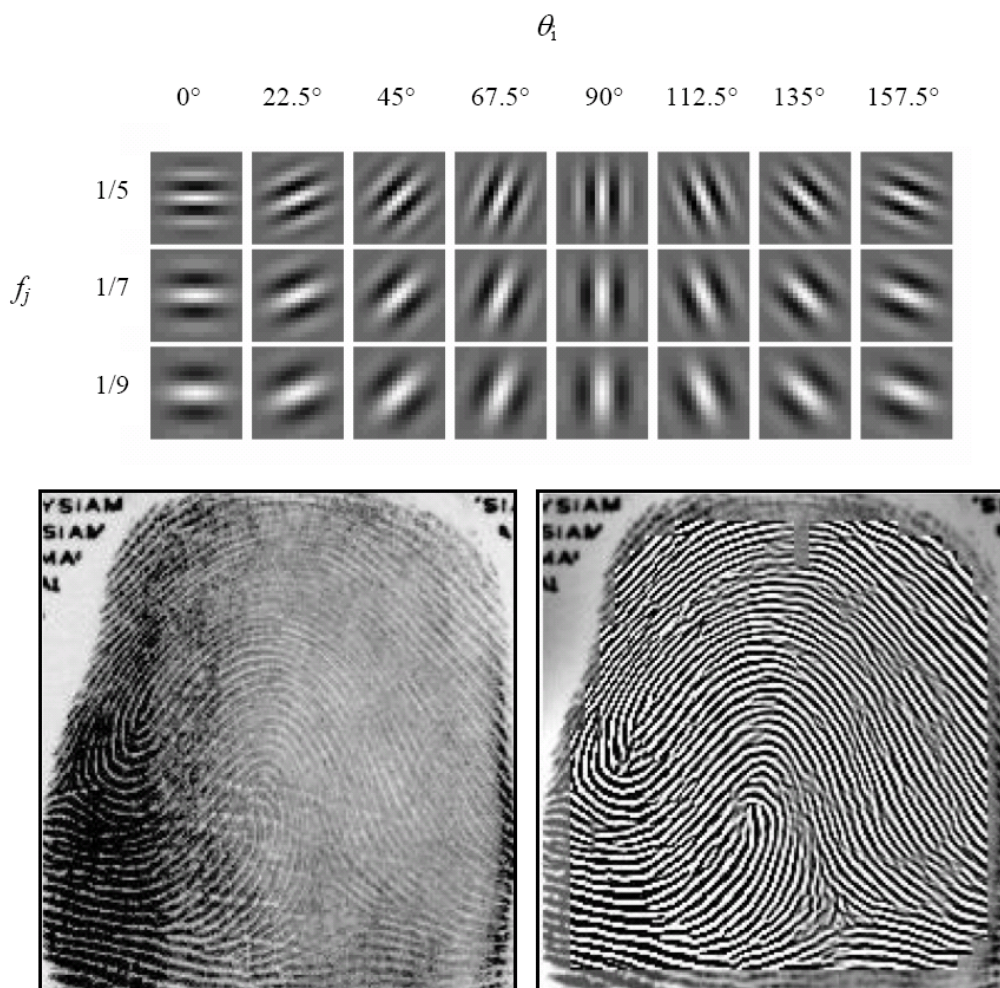
Detekce markantů z otisku prstu a algoritmy, které se k tomu používají velmi závisí na kvalitě obrázku pořízeného otisku. V ideálním otisku prstu jsou jasně patrné všechny papilární linie a v prostoru mezi nimi nejsou žádné nepřesnosti. V takovémto případě lze jednotlivé papilární linie snadno prozkoumat a detekovat všechny markanty.

Ve skutečnosti však díky dalším vlivům, jako je povrch místa, kde je otisk zanechán, stavu kůže (zda je vlhká, suchá, pořezaná, atd.), nečistotám na snímači, špatnému tlaku na celý prst při vytváření otisku a přirozeně poškozených papilárních linií na prstech starších a manuálně pracujících lidí je přibližně 10% otisků prstů nízké kvality.

Cílem algoritmů pro zlepšení obrazu je zvýšení čistoty struktury papilárních linií v některých oblastech a vyznačení oblastí, které jsou pro detekci nepoužitelné díky špatné kvalitě. Vstupem takového algoritmu bývá většinou obrázek ve stupni šedi. Výstupem bývá obvykle také obrázek ve stupních šedi, popřípadě černobílý obrázek, v závislosti na použitém algoritmu.

Většina technik na vylepšování obrazu otisku prstu je založena na kontextových filtrech. V obecném filtrování obrazu se většinou použije pouze jeden filtr, který je aplikován na celý obrázek. Při kontextovém filtrování se charakteristiky filtru mění v závislosti na momentálním místě obrázku, který chceme filtrovat. Obvykle je předem připravena sada filtrů, kdy se vždy jeden použije na příslušnou oblast otisku prstu. Kontext, podle kterého vybíráme příslušný filtr, je ve většině případů ovlivněn orientací a hustotou rozložení v daném místě. Za použití odpovídajícího filtru na patřičném místě v závislosti na orientaci a hustotě jsme schopni odstranit šumy z pozadí a zlepšit tak čitelnost otisku při další práci s ním.

Jistá skupina lidí [3] navrhla efektivní metodu založenou na Gaborových filtrech. Gaborovy filtry mají jak frekvenční (hustota rozložení) vlastnosti, tak i vlastnosti odpovídající orientaci. Na obrázku 3.6 je ukázka sady 24 filtrů, které jsou vybrány tak, aby pokryly celé spektrum úhlu natočení a do jisté míry i hustotu rozložení papilárních linií. Ve spodní části obrázku vlevo je pak obrázek otisku prstu, na který se tyto filtry aplikují a výsledek této operace je zobrazen na pravé části obrázku. Na tomto výsledku je vidět, jak tošlo ke zvýraznění jednotlivých papilárních linií a prostor mezi nimi byl zase zesvětlen. Tím se tento otisk stává daleko lépe čitelnější a rozpoznatelný pro algoritmické zpracování.

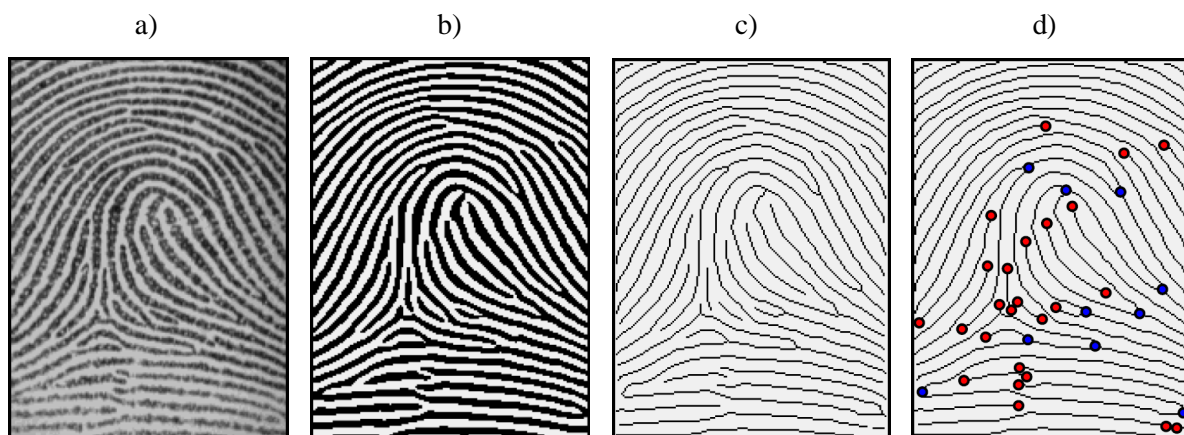


Obrázek 3.6: Užití gaborových filtrů pro binarizaci

## 3.6 Detekce markantů

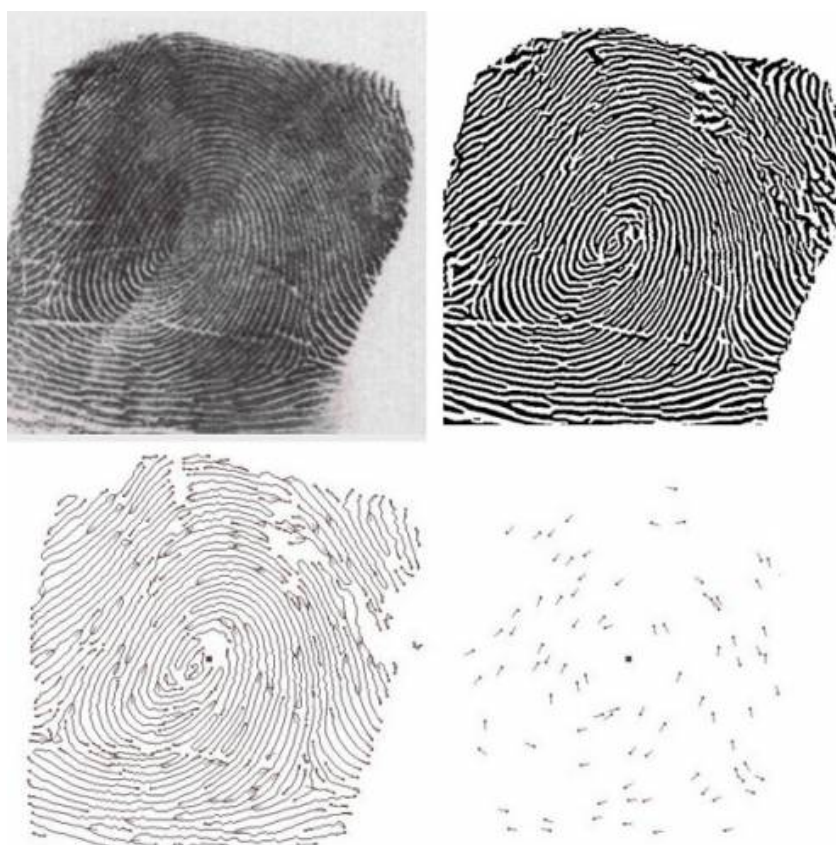
Většina různých algoritmů k samotné detekci markantů nepracuje s obrázkem ve stupních šedi, ale potřebuje ho mít černobílý (binární). Některým binarizačním procesům velmi usnadňuje práci předchozí vylepšení obrazu pomocí předchozích algoritmů. Vylepšovací algoritmy většinou rovnou vrací binární výstup, navíc rozdíl mezi vylepšeným obrazem a binarizovaným obrazem je většinou malý.

Binární obrazy jsou většinou během binarizačního procesu upraveny tak, že jsou papilární linie převedeny do linkové podoby, kdy je pro reprezentaci použita pouze tenká čára. Toto ve finále rozpoznávacím algoritmům usnadní detekci markantů, protože je snazší detekovat rozdvojení nebo konec linky, která je tenká. U neupraveného otisku by toto rozpoznání bylo složitější kvůli kontrole šířko okolí právě zpracovávaného bodu.



**Obrázek 3.7: Úpravy otisku pro detekci markantů**

Na obrázku 3.7 je postupně zleva doprava vidět přibližný způsob úprav, které se aplikují na otisk prstu. Na obrázku a) je otisk ve stupních šedi bez žádných šumů a bez rušivého pozadí. Tento otisk je vygenerován metodou SFinGe. Tento otisk byl dále binarizačními algoritmy převeden do černobílé podoby, což je znázorněno na obrázku b). Ty papilární linie se převedli do linkové podoby zobrazené na obrázku c) a nakonec došlo k detekci markantů na obrázku d), kde červené tečky reprezentují zakončení a modré rozdkvojení.



**Obrázek 3.8: Detekce markantů postupným zpracováním otisku**

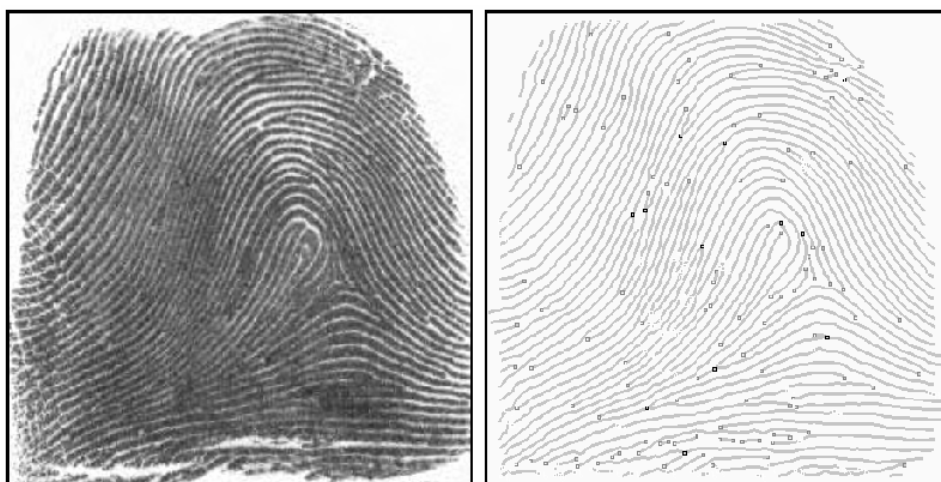


Na obrázku 3.8 je pak zobrazen příklad zpracování a konečné detekce markantů na skutečném otisku prstu [11].

Někteří lidé prosazují detekci markantů, která se provádí přímo z otisku, který je ve stupních šedi a to z následujících důvodů:

- během binarizace může dojít ke ztrátě informací
- binarizace a převod na jednobodové papilární linie zabere nějaký čas; také to může mít za následek chybné detekování markantů
- bez použití vylepšovacích algoritmů vrací většina binarizačních technik neuspokojivé výsledky, když jsou binarizační techniky použity na otisk v nízké kvalitě

Maio a Maltoni navrhli detekční metodu, která pracuje přímo s obrázkem ve stupních šedi. Její základní ideou je, že se postupně „proplouvá“ papilárními liniemi v závislosti na momentální orientaci papilární linie v daném místě. Výsledek se ještě nadále upravuje, aby došlo k odstranění sporně detekovaných markantů.



**Obrázek 3.9: Přímá detekce markantů z obrázku ve stupních šedi**

Ukázka výsledků tohoto algoritmu je vyobrazena na obrázku 3.9. Na tomto obrázku je vidět, že tato metoda taky splňuje svůj účel a vrací poměrně korektní výsledky.

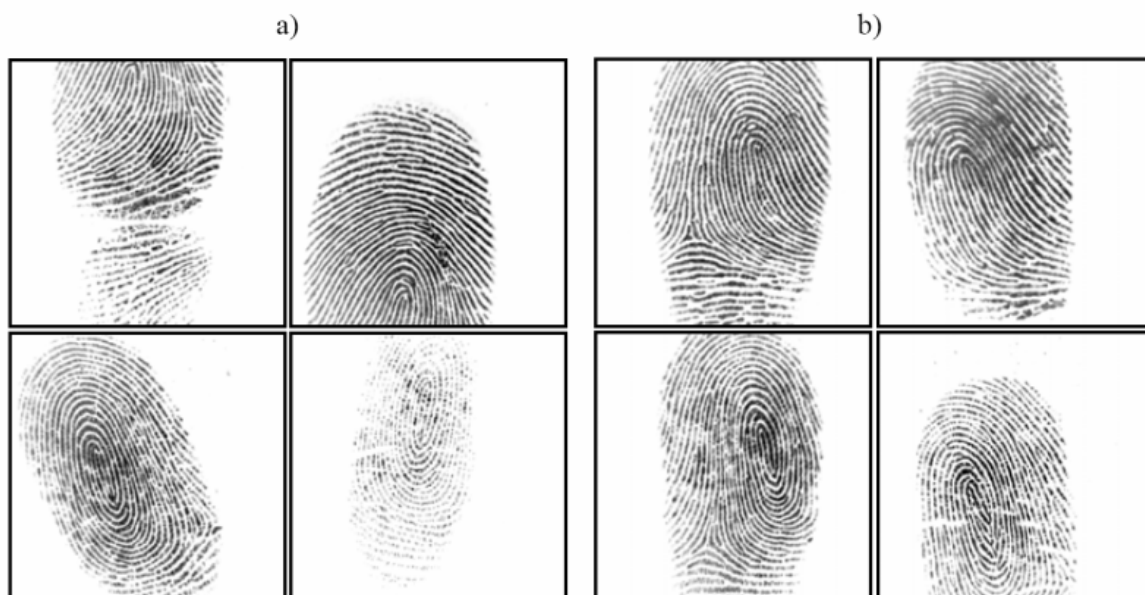
Nelze jednoznačně určit, který způsob a která metoda je pro zpracování otisků nejlepší, protože každá má svoje výhody a nevýhody. Je však jisté, že čím kvalitněji jsou otisky pořízeny, tím menší problémy jsou potom s detekcí markantů u obou metod.

## 3.7 Porovnání otisků

Porovnávání kvalitně pořízených otisků prstů s malými variacemi vlastní třídy není obtížné a dokáže to každý rozumný algoritmus.

Opravdovou výzvou je porovnávání vzorků, kdy nastane několik nepříjemných okolností jako:

- Velké posunutí a natočení: velký posun prstu a jeho natočení má za následek to, že část otisku prstu se nachází mimo pole senzoru, tím pádem máme k porovnání oproti otisku, s kterým porovnáваме, menší plochu
- Nelineární deformace: když se snímá otisk, tak ve skutečnosti snímáme trojrozměrný objekt dvojrozměrným senzorem. Díky plastičnosti prstu může dojít k jinému rozložení na snímači
- Rozdílný tlak při stisku a stav kůže: při příliš silném tlaku může dojít ke sloučení některých papilárních linií, naopak při příliš malém tlaku nedojde ke snímnutí části prstu a tím pádem nejsme přesně schopni rozeznat markanty. Na to má také vliv, zda je kůže suchá nebo mokrá. V případě poraněného prstu může dojít k rozdílným detekcím markantů
- Chyba při výpočtu: algoritmy nejsou vždy bezchybné a v některých případech může dojít k chybnému rozpoznání. Například v málo kvalitním otisku prstu dojde k detekci více nesprávných markantů, což má při porovnávání otisků nepříznivý vliv a je obtížnější poznat shodu



Obrázek 3.10: Porovnání otisků



Na obrázku 3.10 v levé čtveřici otisků jsou na řádku vždy stejné otisky, avšak v různé kvalitě. Tyto otisky byly rozpoznány jako neshodné. Na pravé straně obrázku je další čtveřice, kdy jsou na řádcích naopak rozdílné otisky, avšak poměrně podobných tvarů. Tyto otisky byly zase rozpoznány jako shodné.

Pro porovnávání otisků prstů existuje mnoho algoritmů. Tyto algoritmy mohou být shrnuty do tří hlavních tříd:

- porovnávání založené na korelaci: obrázky dvou otisků prstů jsou položeny přes sebe a je spočítána korelace mezi odpovídajícími pixely pro různé pozice (různé posunutí a natočení)
- porovnávání založené na markantech: z obou otisků jsou zjištěny množiny markantů. Tyto množiny se porovnávají a hledá se většinou určitý počet markantů nalézajících se na stejné pozici
- porovnávání založené na tvarech papilárních linií: porovnávají se tvary papilárních linií zahrnující i jejich hustotu rozložení

## 4 Generování otisků prstů

Generování otisku prstu je svým způsobem opačná metoda oproti rozpoznávání otisků prstů. Pouze jsme oproštěni od rozpoznávacích metod a vylepšování obrazu, protože tvoříme otisk vlastní, čili může být svým způsobem dokonalý, co se týče čitelnosti papilárních linií.

Důvodů pro generování otisků prstů je mnoho. Hlavním důvodem je snadný přístup k otiskům prstů, při jejichž pořizování na skutečných lidech bychom strávili dlouhou dobu. Můžeme si tedy sami vygenerovat vlastní databázi otisků a na ní například testovat rozlišovací algoritmy pro porovnávání otisků a tím tyto algoritmy vylepšovat, aby pracovaly s co nejmenší chybovostí.

### 4.1 Metoda SFinGe

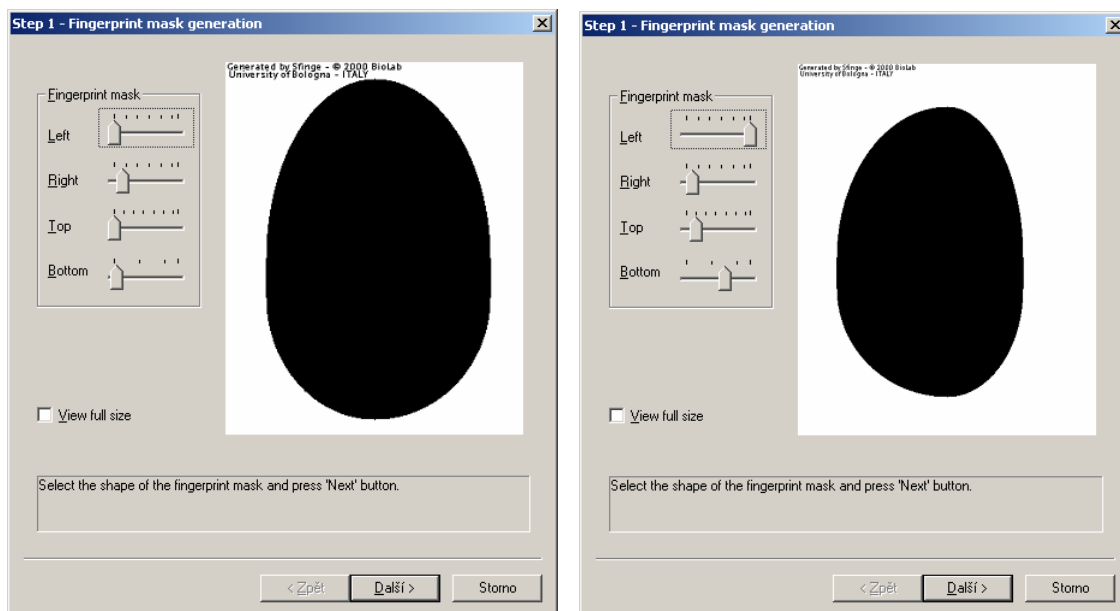
Základní idea metody SFinGe je jednoduchá, generuje se odděleně tvar otisku, pole orientací a mapa rozložení hustoty. Tyto tři rysy jsou zkombinovány k získání vzorku otisku prstu.

Aplikace využívající tuto metodu je vyvíjena na italské univerzitě v Boloni, kde se i v dnešní době pracuje na novější verzi aplikace.

#### 4.1.1 Generování tvaru otisku

V tomto kroku se definuje tvar otisku, v jakém bude generovaný otisk reprezentován.

Tvar je založen na jednoduchém modelu, kdy otisk má elipsoidní tvar a je hodnoty tohoto oválného tvaru se dají nastavovat do všech směrů pro nastavení správné velikosti a tvaru otisku prstu.



Obrázek 4.1: SFinGe – Definování tvaru otisku

## 4.1.2 Generování pole orientací

Tento krok slouží k definování směru natočení papilárních linií v jednotlivých částech otisku prstu.

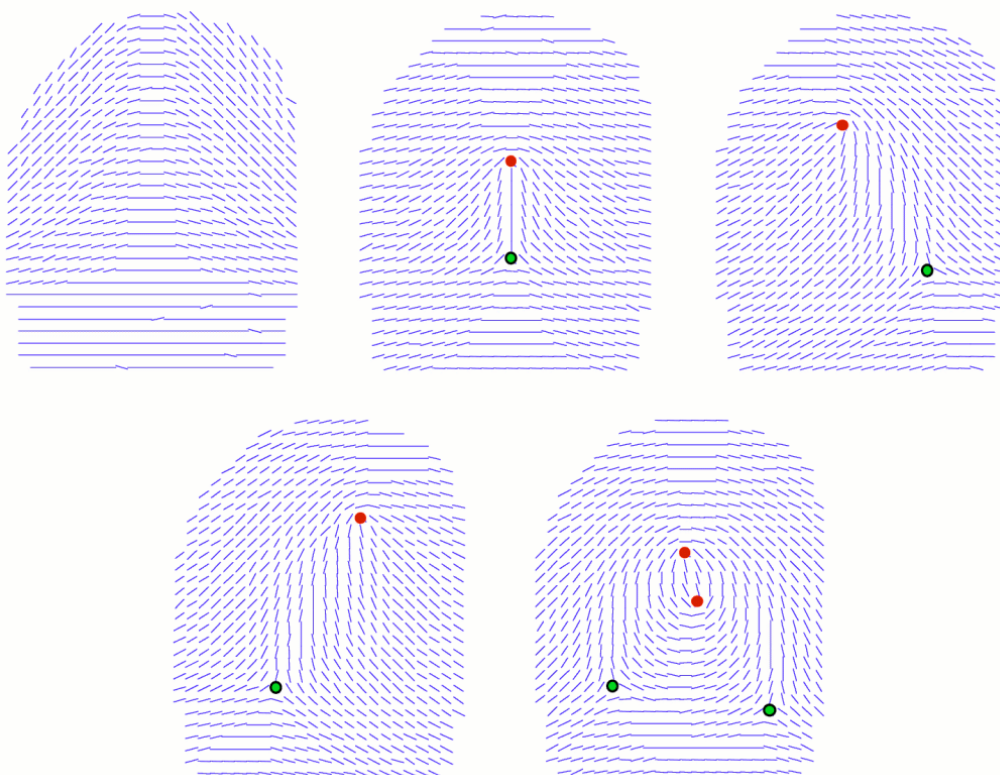
Orientační model navržený Sherlockem a Monroem umožňuje výpočet pole orientací pouze z pozic jádra a delt. V tomto modelu je obraz otisku reprezentován na komplexní ploše.

Nechť  $c_i$  jsou souřadnice jádra, kdy  $i=1..n_c$  a  $d_i$  jsou souřadnice delt, kdy  $i=1..n_d$ . Celkový počet jader a delt samozřejmě není libovolný. Orientace  $O$  každého bodu  $z$  je spočítána takto [3]:

$$O(z) = O_0 + \frac{1}{2} \left[ \sum_{i=1}^{n_d} \arg(z - d_i) - \sum_{i=1}^{n_c} \arg(z - c_i) \right] \quad (4.1)$$

kde  $O_0$  je orientace pozadí. Pokud tedy budeme brát nijak neotočený otisk, bude  $O_0=0$ . Funkce  $\arg(z)$  vrací argument komplexního čísla  $z$ .

Na obrázku 4.2 je vidět příklad vygenerovaných otisků a jejich polí orientací dle různých základních typů



Obrázek 4.2: SFinGe – Ukázka pole orientací podle třídy otisku

### 4.1.3 Generování rozložení hustoty

Pokud se pozorněji zadíváme na jakýkoli otisk prstu, zjistíme, že hustota rozložení papilárních linií na něm není stejná a tudíž vzdálenosti mezi papilárními liniemi se na otisku různí podle polohy.

Po zkoumání obrázků několika otisků prstů se došlo k závěru, že generování mapy rozložení hustoty nelze ponechat zcela náhodě. Došlo se ke zjištění, že většinou u nejsevernějšího jádra a v oblastech níže od nejjižnější delty je hustota rozložení menší než na zbytku plochy otisku prstu. Metoda SFinGe proto používá konstantně navrženou hustotu rozložení, proto ji pro každý otisk nemusí počítat.

### 4.1.4 Generování vzorku otisku prstu

Pokud již máme na vstupu tvar otisku, jeho pole orientací a rozložení hustoty, může se přejít ke generování vzorku papilárních linií, což ovšem není jednoduchý úkol.

K úspěšnému generování je třeba stanovit množinu vytvářejících pravidel, abychom se vypořádali s komplexností papilárních linií v otisku. Tato metoda je velmi jednoduchá, ale zároveň překvapivě účinná. Postupným vylepšováním původního obrazu (obsahujícího jeden nebo více izolovaných černých bodů) pomocí Gaborova filtru za přispění pole orientací a hustoty rozložení vznikne konzistentní a velmi realistický vzorek otisku. Markanty různých typů (zakončení, rozdělení, ostrovy, body, atd.) jsou automaticky generovány na náhodných pozicích. Filtr je získán z Gaussova rozložení a cosinové rovinné vlny; korekční člen nám zaručuje, že je filtr bez stejnosměrné složky [3].

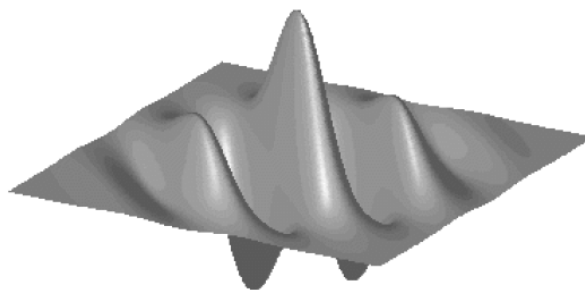
$$f(v) = \frac{1}{S^2} e^{-\frac{\|v\|^2}{2S^2}} \left( \cos(k \cdot v) - e^{-\frac{(s\|k\|)^2}{2}} \right) \quad (4.2)$$

kde  $\sigma$  je parametrem Gaussova rozložení a  $k$  je vlnovým vektorem rovinné vlny. Parametry  $\sigma$  a  $k$  jsou upravovány pomocí pole orientací a rozložení hustoty. Necht'  $z$  je bod v obraze, na kterém má být filtr aplikován, potom vektor  $k = [k_x, k_y]^T$  závisí na výsledku dvou rovnic:

$$D(z) = \sqrt{k_x^2 + k_y^2} \quad (4.3)$$

$$\tan(O(z)) = -\frac{k_x}{k_y} \quad (4.4)$$

Na obrázku 4.3 je graficky reprezentován filtr vytvořený rovnicí 4.2.



**Obrázek 4.3: SFinGe – Gaborův filtr používaný v této metodě**

### **4.1.5 Další případné úpravy**

Když je nyní otisk vygenerován, je čistý a bez žádných nepřesností, nečistot, a vypadá, jakoby tlak při stisku prstu byl naprosto stejnoměrný.

Pro větší realističnost se proto obrázek otisku ještě upravuje, aby působil realističtěji a aby generované otisky bylo možné předložit rozpoznávacím algoritmům kvůli jejich testování. Proto se přidává simulace nestejnoměrného tlaku při tisknutí prstu na senzor, přidává se šum, nepřesnosti a jizvy. Simulovat se dá i vlhkost prstu při pořizování otisku. Nakonec se dá přidat i nějaké předem definované pozadí pod otisk prstu.

### **4.1.6 Příklady výsledků metody SFinGe**

Jak je patrné z obrázku 4.4, otisky vygenerované touto metodou vypadají velmi realisticky [3].



**Obrázek 4.4: SFinGe – Vygenerované otisky prstů**

## 4.2 Vlastní návrh řešení projektu a jeho algoritmus

Cílem našeho projektu bude algoritmus, který zužitkuje předem získané znalosti o otiscích prstů, jejich rozpoznávání a metodách užitých v metodě SFinGe.

### 4.2.1 Uvedení do problematiky vlastního řešení

Pokud tedy chceme generovat otisky prstů, bude rozumné ubírat se přibližně směrem, kterým se již dala metoda SFinGe a postupovat v základních krocích, které jsou již navrženy a svým způsobem ověřeny.

Naším cílem je tedy z generovaných nebo námi zadaných markantů vytvoření vzorku otisku s vyobrazením papilárních linií. Z tohoto vzoru už by nemělo být obtížné aplikováním specifických filtrů vytvořit otisk odpovídající svým vzhledem přibližně otisku skutečnému. Dalšími přídatnými filtry by se dalo simulovat zjizvení otisku, popřípadě simulovat různé povrchy, na kterých byl otisk zanechán.

U metody SFinGe se na vstupu nezadávají pozice markantů nějakého konkrétního otisku prstu, ale dochází ke kompletnímu generování otisku dle zadaného tvaru prstu a třídy otisku, který byl vybrán. Tento nepatrný detail má ovšem velký vliv na samotné řešení projektu a jeho obtížnost, protože v tomto případě se zná tvar a natočení otisku a singularity v něm, které ovlivňují a svým způsobem definují hlavní orientaci jednotlivých papilárních linií. Tím tedy máme zajištěn snadnější výpočet pole orientací a přesnější definici samotného otisku prstu.

Při generování otisku podle daných markantů však typ otisku ani pozice singularit neznáme a patrně by nebylo nejjednodušší tyto věci definovat, protože nemusíme vědět, podle čeho byly markanty vytvořeny, v jaké jsou skutečné poloze vůči otisku prstu a jaký měl tento otisk natočení. Vygenerováním otisku podle uživatelem zadaných singularit by jednoduše mohlo snadno dojít k vytvoření pole orientací, které by nekorespondovalo se všemy markanty a jejich směru udávající orientaci papilární linie. Metoda jako taková se tedy bude muset upravit, abychom dosáhli uspokojivých výsledků.

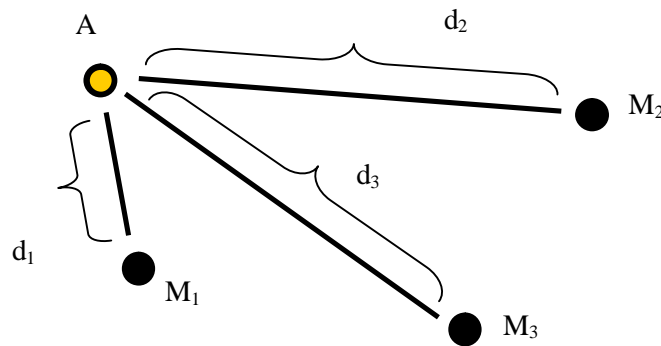
### 4.2.2 Pole orientací a jeho generování

Pole orientací je jedna z nejdůležitějších částí pro vygenerování otisku, protože toto pole představuje konkrétní popis papilárních linií a jejich natočení v každém bodě obrázku, kde se otisk generuje. Při zpracovávání otisků se někdy z papilárních linií detekuje a sestavuje pole orientací, z toho vyplývá, že by měl jít postup rovněž obrátit a z pole orientací opět sestavit přibližně totožný otisk.

Pokud jsme se tedy rozhodli toto pole generovat a není na místě užití rozmístění singularit a jejich vliv na orientaci, musí se pole vytvořit na základě jiných údajů. Na vstupu máme pouze množinu markantů (jejich pozici a natočení na dané pozici), čehož se dá také využít.

Pole orientací představuje pole o rozměrech obrázku, na který budeme otisk generovat a pro každý pixel tohoto pole se musí spočítat orientace (směr, kterým se ubírá papilární linie). Z již zadáných markantů se zanesou příslušné hodnoty natočení do pole orientací a zbylé hodnoty pole (pro všechny ostatní pixely) se díky těmto pevně daným bodům musí dopočítat. Dle úvahy je logické, že na právě počítanou hodnotu orientace nějakého pixelu bude mít největší vliv markant, který je tomuto bodu nejbližší. V případě, že nejbližší budou markanty dva a navíc stejně vzdálené, výsledné natočení ovlivňují oba markanty stejně.

Za tímto účelem musíme zjistit vzdálenosti ke všem markantům od počítaného pixelu a váhově je ohodnotit, abychom získali ke každému markantu a jeho úhlu natočení také váhu markantu, jakou daný pixel ovlivňuje.



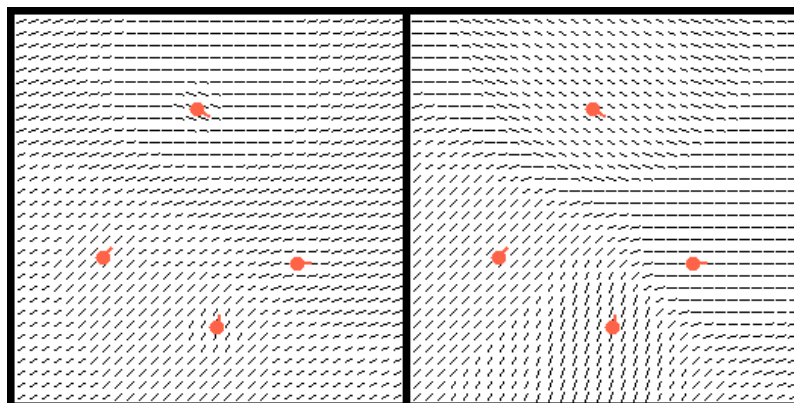
**Obrázek 4.5: Výpočet orientace**

Na obrázku 4.5 je patrné, že největší vliv na výslednou orientaci pixelu A bude mít první markant  $M_1$ , protože je nejbližší.

Matematicky má ovšem tento markant vzdálenost od pixelu nejmenší, proto se výpočet vah bude provádět s převrácenými hodnotami vzdáleností. Musíme tedy zjištěné vzdálenosti normalizovat do použitelné podoby, viz následující vzorec 4.5, který zajistí pro každý markant již normalizovanou hodnotu vzdálenosti, představující přímo váhu, která nám umožní počítat se všemi markanty najednou.

$$w_{norm} = \frac{\frac{1}{vzdálenost^p}}{\sum \frac{1}{vzdálenost^p}} \quad (4.5)$$

Hodnota  $P$  v mocnině zde představuje sílu vzdálenosti, kdy váhové rozložení nemusí být obyčejné lineární rozložení. Tato síla dopomůže celkový výpočet ovlivnit tak, že čím dále markanty jsou, tím menší váhu mají a to ne podle lineárního výpočtu, ale třeba v závislosti na kvadrantu vzdálenosti. Na obrázku 4.6 navlevo je použita první mocnina a napravo mocnina čtvrtá. Na pravé části obrázku je vidět, jak markanty kolem sebe rozšiřují svoje natočení s daleko větší razancí, než markanty v levé části obrázku. Tím dojde k rozdílnému pojetí celého pole orientací a vzájemných závislostí mezi jednotlivými markanty.



**Obrázek 4.6: Rozdíl síly váhování**

Díky takto vytvořenému poli orientací už máme lepší představu o směrech, kterými se můžou ubírat papilární linie tak, aby na pozicích markantů měly směr korespondující s natočením markantu.

V algoritmu se neberou v úvahu singularity, i když teoreticky by se do celkového výpočtu orientací daly zahrnout také. Tak by ovšem vyvstala spousta dalších problémů, protože singularity ovlivňují pole orientací jako celek a tím by bylo obtížnější udržet na všech pozicích markantů správný směr papilární linie. Například za předpokladu, že bychom měli sadu markantů získanou z nějakého skutečného otisku prstu, nemáme záruku, že všechny markanty z tohoto otisku byly rozpoznány přesně a že jejich úhel je detekován správně. Pokud bychom pak měli další algoritmus na zařazení singularit do nově generovaného otisku ze zadaných markantů, musel by tento algoritmus umět počítat s jistou chybou a takové nepřesnosti předpokládat. Práce s otisky však není vždy bezchybná, protože při snímání nebo skenování otisků může dojít k nepřesnostem a dokonce může být z otisku rozpoznán markant na místě, kde žádný markant není. Proto by bylo neefektivní počítat s tím, že všechny markanty jsou zadány bez chyby.

Na tomto případě je krásně patrné, čím je algoritmus pro generování otisků prstů z předem daných markantů obtížnější oproti metodě SFinGe a jejího generování pole orientací na základě singularit, kdy se markanty vytváří náhodně až při samotném generování vzoru papilárních linií.



Naše pole orientací je však dostatečné a použitelné, protože svým způsobem také simuluje směr papilárních linií. Tyto směry sice nekorespondují s žádným typem otisku prstu, zato je jistota, že u všech markantů bude zachována spojitost s papilární linií a úhlem markantu na jeho souřadnicích.

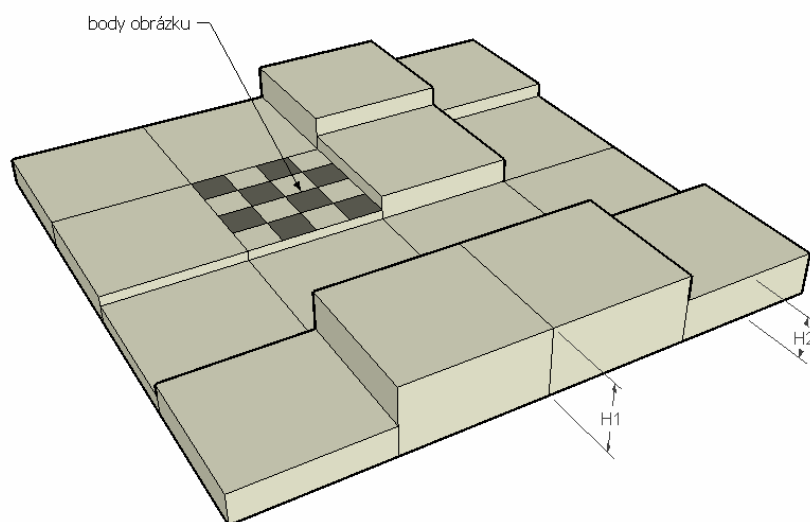
Natočení markantů může být v rozmezí  $360^\circ$ , protože když máme například markant vidlice, kdy se papilární linie rozdvouje, potřebujeme znát směr rozdvojení v rozmezí celého kruhu. U pole orientací však všeobecně víme, že nám stačí hodnoty v rozmezí 0 až 179 stupňů, jak jsme se dozvěděli v kapitole 3.2.1. Proto i naše pole orientací bude mít takovýto rozsah hodnot.

### 4.2.3 Rozložení hustoty

Rozložení hustoty papilárních linií je nedílnou součástí pro generování vzorku otisku prstu. Pro náš algoritmus to však není jedna z důležitých částí, protože od vygenerovaného vzorku otisku je požadováno správné rozložení markantů, které byly na vstupu a případné zhušťování papilárních linií na to velký vliv mít nebude.

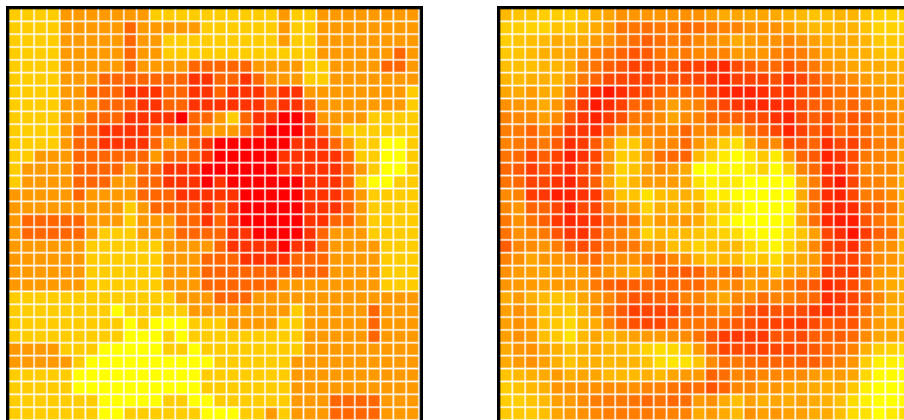
I když však toto není pro náš algoritmus důležitá část, přesto se může stát jeho součástí. V metodě SFinGe je používáno konstantní rozložení hustoty získané na základě zkušeností a pozorování otisků prstů. Jelikož je u této metody znám tvar otisku prstu a jsou známy pozice špičky prstu, jeho střed a okraje, mohou si takové konstantní rozložení dovolit. V našem případě bude nejrozumnější toto rozložení dělat proměnné, aby šlo případně měnit a díky tomu i rozšířit celkové možnosti celého projektu.

K vytvoření rozložení hustoty stačí udělat pole o rozměrech korespondujících s velikostí obrázku, na který bude otisk generován a v tomto poli definovat pro všechna místa hustotu rozložení nějakou proměnnou hodnotou.



**Obrázek 4.7: Reprezentace rozložení hustoty**

Na hustotu nemusí být brán takový zřetel s přesností, co se detailů týče a nemusí být nastavitelná pro každý pixel zvlášť. Tyto proměnlivé hodnoty se dají pro představu reprezentovat i graficky, jako je tomu na obrázku 4.7, kde je pro reprezentaci zvolena proměnlivá výška v závislosti na míře hustoty. Vyšší bloky mohou například představovat větší hustotu papilárních linií v daném místě a u bloků nižších to bude naopak. Jeden blok se může skládat z většího množství pixelů, čímž se reprezentuje dříve zmíněná nepotřebnost informace o hustotě na každém pixelu zvlášť. Pro reprezentaci rozložení hustoty je běžnější užití reprezentace podle odstínu barev, avšak pro lepší představu je myslím výšková reprezentace názornější. V aplikacích je však lepší užití přepočtu na reprezentaci barvami, protože lze díky tomu okamžitě danou hustotu znázornit a po případné editaci této hustoty hned vidět změny. Reprezentace v barevném provedení a je ukázána na obrázku 4.8. Tyto hustoty rozložení byly vytvořeny ve výsledné aplikaci tohoto projektu.



**Obrázek 4.8: Reprezentace rozložení hustoty pomocí barev**

Při samotném generování otisku prstu pak už jen stačí toto rozložení hustoty zohlednit a papilární linie generovat v závislosti hodnot hustoty.

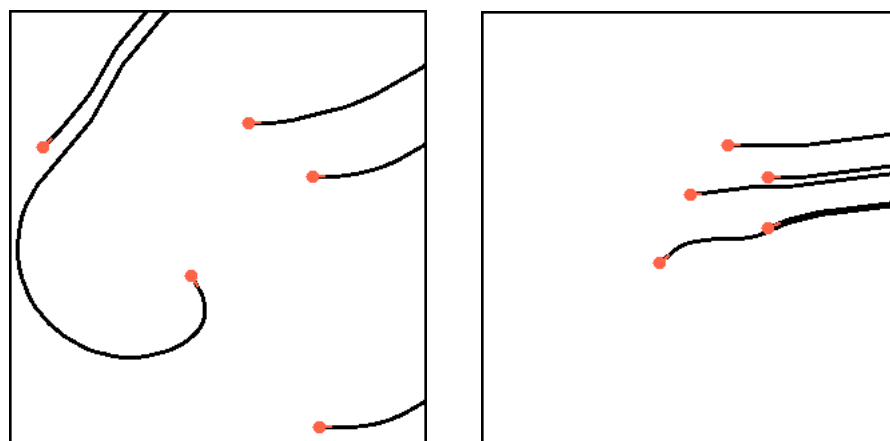
#### **4.2.4 Možná řešení generování vzoru**

U metody SFinGe pro generování otisku stačí tvar otisku, pole orientací a rozložení hustoty. Tyto údaje svým způsobem máme také, můžeme tedy přistoupit k té nejdůležitější části a to je samotné generování papilárních linií. Způsobů pro generování může být více.

##### **4.2.4.1 Výpočet papilárních linií**

Jedním z možných řešení, které člověka napadne jako první je, že pomocí získaných dat by se dalo přesně vypočítat průchody papilárních linií od zadaných markantů až mimo plochu, kterou pro otisk potřebujeme.

Tento postup má velmi mnoho výhod, ale také mnoho úskalí a nepříjemností, která řešení problematiky tímto způsobem značně ztěžují. Pokud bychom například vycházeli z každého markantu a ubírali se směrem, který nám sdělí pole orientací, měli bychom ve výsledku množinu bodů reprezentující pokaždé jednu křivku a tyto křivky by ve finále vylézaly z kreslicí oblasti ven. Na první pohled to vypadá dobře, ale neztotožňuje se to příliš s vlastnostmi a vzhledem papilárních linií. V tomto případě by se totiž křivky ve většině případů nakonec sbíhaly ve shluk čar a s velkou pravděpodobností by se také překrývaly, takže by se tyto papilární linie od sebe nedaly rozeznat. Ukázka takového případu je zobrazena na obrázku 4.9.



**Obrázek 4.9: Zobrazení křivek**

U menšího počtu markantů se může zdát skoro vše v pořádku, další souběžné papilární linie by se daly také teoreticky dopočítat, ale největším problémem je, že křivky nevytváří ucelený obrazec čar. Na obrázku 4.9 vpravo je vidět, jak si křivky hledí svého směru podle pole orientací. Aby tedy tato metoda byla účinná, muselo by se vyřešit slučování samotných křivek ve složitější struktury.

V případě vyřešení problémů tohoto způsobu vykreslování bychom získali velmi mocný nástroj pro kresbu papilárních linií, protože samotný přepočet křivek by nemusel být výpočetně náročný. Po případné změně pozice nějakého markantu bychom během chvilky mohli vidět, jak algoritmus zareaguje a jak to ovlivní celkový vzhled obrazu otisku.

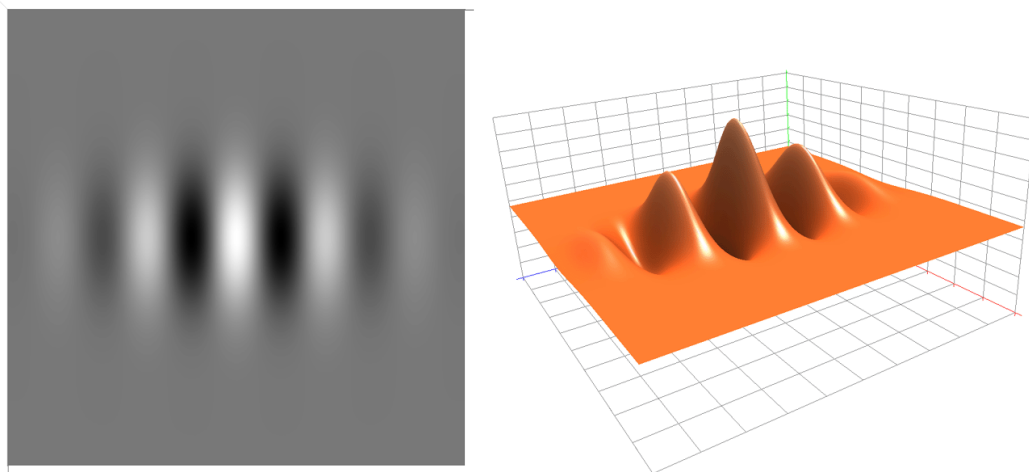
#### **4.2.4.2 Aplikace filtru a úprava obrázku**

Jako další metoda pro vytvoření papilárních linií připadá stejný způsob, který používá metoda SFinGe, ale s tím rozdílem, že se zohlední již existující markanty. Tato metoda využívá aplikování gaborova filtru na výstupní obrázek a dopočet jednotlivých pixelů obrázku. Podrobněji tento způsob generování bude popsán v kapitole 4.2.6.

## 4.2.5 Gaborovy filtry

Gaborovy filtry jsou součástí algoritmu pro generování vzorku otisku prstu, proto v této kapitole bude více přiblížen jejich matematický základ.

Filtry jako takové mají mnoho uplatnění, nás však nejvíce zajímá uplatnění ve filtrování obrazu. Obdobné filtry se používají i při binarizaci, kdy se ze snímnutého obrazu otisku prstu obraz upravuje pro lepší rozpoznání hran a následnou lepší detekci markantů.



**Obrázek 4.10: Reprezentace gaborova filtru**

Na obrázku 4.10 vlevo je ukázka vzhledu 2D filtru převedeného do stupně šedi, kde nejsvětlejší místo uprostřed koresponduje s nejvyšším místem na pravé části obrázku [8]. Na binarizaci obrázku s otiskem prstu je takový filtr ideální, protože filtr je díky svým parametrům rovnice nastavitelný dle požadavků a umožní eliminovat z obrazu nežádoucí šumy a zaručí zvýraznění papilárních linií.

Následující vzorec reprezentuje matematické řešení takového filtru pro jeho rekonstrukci a jednotlivé parametry vzorce jsou vysvětleny dál [7]:

$$g(x, y) = \exp\left(-\frac{x_g^2 + g^2 y_g^2}{2s^2}\right) \cdot \cos\left(2p \frac{x_g}{l} + j\right) \quad (4.6)$$

$$\begin{aligned} x_g &= x \cdot \cos q + y \cdot \sin q \\ y_g &= -x \cdot \sin q + y \cdot \cos q \end{aligned} \quad (4.7)$$

### 4.2.5.1 Vlnová délka (wavelength) – $\lambda$ (lambda)

Specifikuje hodnotu cosinu gaborovy funkce.

Hodnoty by měly být reálná čísla větší nebo rovna 2. Hodnota 2 by však neměla být použita v kombinaci s fázovým posunem  $\varphi = -90$  nebo  $\varphi = 90$ , protože v těchto případech by byl výsledek gaborovy funkce nežádoucí a to díky výpočtu funkce cosinu.



**Obrázek 4.11: Gaborův filtr - vlnová délka**

Na obrázku 4.11 je ukázka filtru pro hodnoty 5, 10 a 15 vlnové délky.

#### 4.2.5.2 Orientace – $\theta$ (theta)

Tento parametr určuje natočení souběžných čar gaborovy funkce. Hodnota představuje úhel a může být v rozmezí od 0 do 360.



**Obrázek 4.12: Gaborův filtr - orientace**

Na obrázku 4.12 je ukázka pro hodnoty 0, 45 a 90.

#### 4.2.5.3 Fázový posun – $\varphi$ (fi)

Tento parametr ovlivňuje hodnotu cosinové části gaborovy funkce.

Hodnoty mohou být od -180 do 180.

#### 4.2.5.4 Prostorový poměr – $\gamma$ (gamma)

Tento parametr specifikuje elipticitu gaborovy funkce.



**Obrázek 4.13: Gaborův filtr – prostorový poměr**

Na obrázku 4.13 je vlevo hodnota poměru 0,5, napravo 1. V pravé části obrázku vidíme, že pro hodnotu 1 je výsledek kruhového tvaru.

## 4.2.6 Generování papilárních linií

Pro řešení projektu jsem zvolil užití gaborova filtru na výstupní obraz, kdy se tento filtr aplikuje na jednotlivé pixely, které se tímto dopočítávají. Pro filtr pro výslednou aplikaci jsem zvolil rozlišení 15x15, což by mělo pro generování stačit.

Metoda SFinGe má v tomto případě velkou výhodu, protože nemusí vytvářet specifické markanty na přesně daných souřadnicích, ale nechá to na náhodě. Kdybychom začali generovat otisk stejně, kdy bychom na obrázek vložily několik počátků (v algoritmu metody SFinGe prezentované černou tečkou), došlo by také k vygenerování vzorku otisku prstu na základě pole orientací, avšak pozice markantů by v žádném případě nekorespondovala se vstupními hodnotami.

Princip užití filtru pro generování je ve své podstatě velmi jednoduchý. Máme sadu filtrů pro různé úhly a případně i hustotu rozložení. Vždy se vybere filtr, který je pro nás v dané chvíli a pro daný pixel nejvhodnější a podle vzorce 4.8 se vypočítá výsledná hodnota barvy pixelu. Výsledek není samozřejmě hned hodnota RGB, avšak hodnoty se pohybují od  $-MAX$  do  $+MAX$ , což není problém přepočítat do odstínu šedi a reprezentovat tak graficky.

$$z_{(x,y)} = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} x_{(x+i,y+j)} \cdot k_{(i,j)} \quad (4.8)$$

$z$	– výsledná hodnota právě počítaného pixelu
$x,y$	– souřadnice počítaného pixelu
$i, j$	– souřadnice filtru
$k$	– hodnota filtru
$I, J$	– velikost filtru

Filtr se tedy aplikuje na výstupní obrázek, na kterém je umístěno 1..n černých bodů, které zapříčiní expanzi sebe sama do svého okolí a to v závislosti na vizuální podobě filtru. Kvůli tomuto důvodu musíme mít filtrů více, aby bylo pokryto širší spektrum natočení podle pole orientací a na výsledném obrázku generovaný obrázek dostával podobu podle pole orientací. V aplikaci jsem zvolil jak možnost mít předdefinovanou sadu filtrů, tak dopočítání každého filtru pro právě zjišťovaný pixel, aby tento výpočet mohl být přesnější. Předem nadefinovaná sada filtru má nevýhodu v zabírání místa v paměti, kdy čím více filtrů máme, tím větší prostor v paměti potřebujeme. V tomto případě se dá ovšem najít kompromis mezi počtem filtrů a přesností, kterou svým počtem budou ovlivňovat.

Generování každého filtru zvlášť nám pro každý pixel přináší přesné hodnoty filtru, avšak výsledná výpočetní náročnost je tímto ovlivněna.

Způsobů, kterými se filtr bude aplikovat na výstupní obrázek, je víc. Souvisí to tak i trochu s optimalizací a požadavky, které na filtr klademe.

#### **4.2.6.1 Procházení pixel po pixelu s jedním nebo více počátky pro expanzi filtru**

Tento způsob je pro generování otisků nejméně efektivní, protože se musí celý obrázek procházet na několik kroků, aby se postupně vytvořil obrázek na celé ploše výstupu. Kroků musí být více, protože ke změně pixelu dojde pouze tehdy, když se filtr přiloží na část obrázku, kde je aspoň jeden pixel nebílé barvy.

#### **4.2.6.2 Procházení pixel po pixelu s konstantním pozadím výstupu nenulové hodnoty**

Pokud výstupnímu obrázku nastavíme barvu pozadí jinou, než je 0, filtr bude vracet nenulové výsledky hned od začátku a generování započne už od prvního pixelu a na další pixely se bude rozšiřovat stále.

#### **4.2.6.3 Definování jednoho nebo více počátků a expanze do jejich okolí**

Tento způsob generování je nejlepší ve srovnání s předchozími metodami a užívá se při generování otisků u metody SFinGe.

Pokud na výstupním obrázku definujeme jeden nebo více bodů a filtr se bude aplikovat postupně na okolí bodu a se zvětšujícím se poloměrem, dojde k vytváření výsledného otisku okamžitě a bez nutnosti aplikovat filtr na celý obrázek znovu.

V případě, že je počátečních bodů více, je logické, že generované papilární linie v okamžiku, kdy se setkají linie vygenerované z různých bodů počátku, na sebe nebudou navazovat, avšak díky filtru a jeho vlastnostem by mělo dojít k jejich sloučení a tím k ideálnímu výsledku vzhledu papilárních linií.

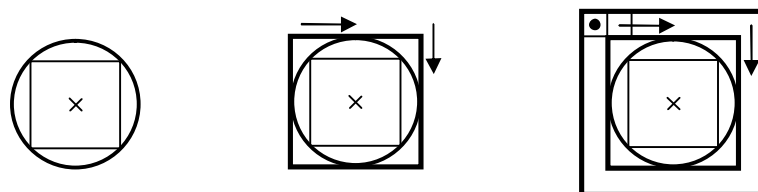
#### **4.2.6.4 Definování celého markantu jako počátku**

Předchozí tři způsoby filtru jsou pro generování otisků prstů s předem definovanými markanty takřka nepoužitelné, protože u nich nejde lehce předvídat, zda se papilární linie vyskytne na místě markantu.

Abychom docílili toho, že po generování otisku budeme mít v obrázku takové markanty na pozicích, které jsme zadali, musíme markanty vykreslit ještě před generováním a zamezit tomu, aby se tyto markanty aplikací filtru změnily nebo dokonce zmizely. Toho docílíme tak, že markant vykreslíme o určitém poloměru kružnice a generovat začneme v okolí markantu od vepsaného čtverce, jak je zobrazeno na obrázku 4.14.

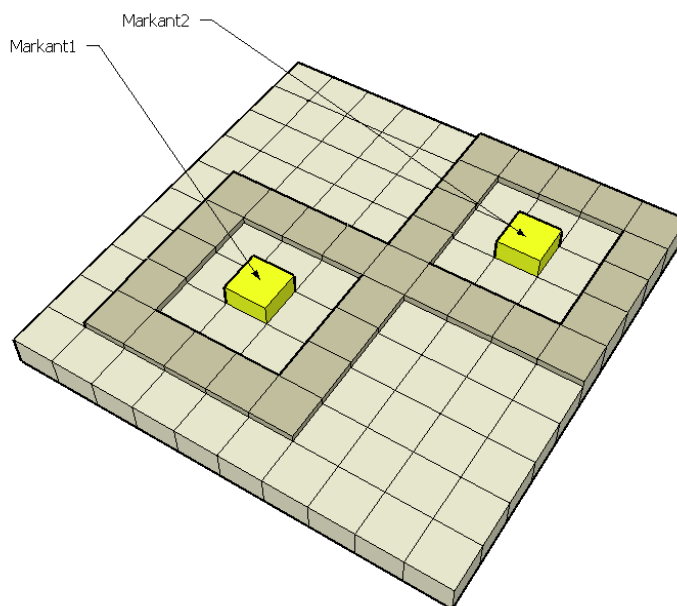
Střed kruhu představuje souřadnici markantu, do kterého se vykreslí markant buď rozdělení nebo za končení a v natočení, jaké bylo zadáno jako vstupní hodnota markantu. Jako varianta pro

možné lepší generování se do kruhu může zakreslit i okolí papilárních linií, které budou daný markant obklopovat.



**Obrázek 4.14: Generování od pozice markantu**

Tímto jsme docílili obdobného řešení jako v předchozí metodě, kdy pro počátek generování, který aktivoval filtr, byla použita černá tečka. V tomto řešení je použit celý markant, avšak aplikace filtru se provede až od určitého okolí markantu. Díky filtru by mělo dojít k expanzi papilárních linií správným směrem a napojením linií na již nakreslenou část markantu v opsané kružnici. Nyní už stačí filtr aplikovat postupně na větší a větší okolí, až dojde k vykreslení celého obrázku. V případě více markantů se pouze nakreslí všechny požadované markanty a filtr se bude současně aplikovat na všechny, což reprezentuje obrázek 4.15.



**Obrázek 4.15: Generování od více markantů najednou**

Žlutá vyvýšená místa reprezentují jednotlivé markanty, kolem kterých je místo, s kterým se algoritmicky nebude nic dělat, aby se docílilo zachování markantu. Tmavší vyvýšená místa už jsou okolí, na kterých se provádí výpočet výsledné barvy. Na obrázku je patrné, že více markantů svou expanzí ovlivňuje i ty pixely, které byly jednou vypočítány. Toto je však to, co potřebujeme, protože



tím dojde k napojení a ucelení papilárních linií z celého obrázku otisku. Už by však nemělo dojít k tomu, aby výpočty pixelů z okolí jednoho markantu začaly ovlivňovat bezprostřední okolí markantu jiného, s kterým se podle algoritmu nepracuje. Okolí všech markantů se tedy do určitých mezí musí označit jako neměnné, aby se na ně filtr neaplikoval.

#### **4.2.7 Selekce zobrazitelné části otisku**

V metodě SFinGe tato část našeho algoritmu nejvíce souvisí s definováním tvaru otisku. Zde však tvar otisku neslouží k výpočtu pole orientací ani ke generování jako takovému.

Bez žádné selekce oblasti, která se má zobrazit, by vygenerovaný výsledek nevypadal jako otisk prstu ale jako změt papilárních linií v celém 2D obrázku. Jako výsledek by to mohlo být dostačující, avšak není na škodu vymezit i nějaký oválný tvar reprezentující svým tvarem prst.

## 5 Řešení projektu

Po návrhu algoritmu pro vlastní generování otisků prstů je potřeba tuto metodu ověřit i v praxi a vytvořit aplikaci, která by s touto metodou pracovala. Programovacích jazyků je velké množství. Ve většině případů se liší hlavně syntaxí, ale základ mají velmi podobný.

Pokud chceme pomocí aplikace zkoumat nějakou problematiku, je potřeba, aby vyvíjená aplikace byla dostatečně interaktivní a komunikovala s uživatelem tak, aby mohl uživatel nastavovat a měnit co nejvíce parametrů. Z tohoto důvodu je tedy velmi důležité mít kvalitní grafické uživatelské rozhraní, které tuto komunikaci s uživatelem zprostředkuje.

V tomto projektu se zároveň pracuje i s obrázky, takže v aplikaci budou potřeba knihovny pracující s multimédií, alespoň grafickými.

Algoritmus jako takový se dá realizovat takřka v libovolném programovacím jazyce, na základě dle mého názoru důležitých aspektů vzhledu aplikace jsem se rozhodl pro implementaci v jazyce C#, který využívá prostředí .NET Framework obsahující vše potřebné pro řešení našeho projektu.

### 5.1 Programovací jazyk C# a jeho možnosti a užití v projektu

Programovací jazyk C# patří mezi poměrně nové programovací jazyky. Je to jazyk vyvinutý firmou Microsoft, stejně jako platforma .NET. Syntaxe jazyka C# je založena na syntaxi jazyků C++ a Java, čímž je vlastně nepřímým potomkem jazyka C [9].

C# má v aplikacích pro operační systém Windows široké uplatnění a užívá se pro tvorbu databázových programů, webových aplikací a služeb a formulářových aplikací.

Jak již bylo dříve zmíněno, jazyk C# využívá prostředí .NET a byl vlastně pro tuto platformu přímo vyvinut. Toto prostředí lze spustit na počítačích se systémy Windows 98, Windows 2000 a Windows XP. Samozřejmě i na novém operačním systému firmy Microsoft Windows Vista má toto prostředí plnou podporu a už je zahrnuto v instalaci operačního systému bez nutnosti aktualizování a doinstalovávání do něj. Tím je tedy zaručena jakási kompatibilita mezi operačními systémy. Samozřejmě ale závisí i na způsobu psaní kódu a na přesné verzi platformy .NET na daném počítači. Dnešním standardem je verze 2.0, v operačním systému Windows Vista je už zabudována verze 3.0. Pro zatím běžnou práci je však verze 2.0 dostačující.

Díky platformě .NET Framework má uživatel plně pod kontrolou klasické komponenty z operačního systému Windows, které si dle uvážení a potřeby vytváří a zároveň zpracovává zprávy a události, které vznikají při interakci mezi uživatelem a těmito komponentami. Takto se dá velmi

jednoduchým a rychlým způsobem vytvořit kvalitní grafické uživatelské rozhraní, které bude splňovat naše potřeby.

Jazyk C# je jazykem objektovým. Pro vytváření windowsové aplikace je tedy ideální a pružný, protože s rozumným návrhem programu programátor vytvoří potřebné objekty a pro tyto objekty pak funkce komunikující s okolím. Objektové programování má mnoho výhod a u složitějších programů značně zlehčuje práci, protože programátor pro sebe během vývoje vytváří nové a nové metody, s kterými si ve výsledku psaní programu ulehčuje a celý program se stává přehlednější. V případě že chce pak uživatel změnit algoritmus v metodě, změní se pouze tato metoda a zbytek aplikace už se přizpůsobí velmi snadno. Tedy pokud nejdojde ke změně vstupních a výstupních dat metody, ale zdrojový kód se této změně dá velmi jednoduše uzpůsobit rovněž.

Pro každou důležitou část projektu byl tedy vytvořen objekt (pole orientací, rozložení hustoty, markanty, ...) reprezentující tuto část a k tomuto objektu byly vytvořeny metody dle uvážení a potřeby.

## 5.2 Možnosti vstupu a výstupu

Většina aplikací, se kterými má uživatel pracovat a mají sloužit nějakému účelu, by měly by nějaký výstup, který uživatel po vykonání nějaké činnosti získá. V některých případech je rozumné mít i možnost načítat nějaká vstupní data, ale to už závisí na užití aplikace.

### 5.2.1 Vstup aplikace

Generátor otisků prstů vstup jako načtená data nepotřebuje, protože nastavení hodnot pro generování otisku prstu se děje v samotné aplikaci.

#### 5.2.1.1 Markanty

V aplikaci se však pracuje s markanty a jejich pozicemi v obrázku, v případě potřeby jsem tedy pomyslel i na načítání těchto hodnot z textového souboru.

Markanty musí být uloženy v textovém souboru takového formátu:

Regulární výraz jednoho řádku textového souboru:

`^(D+)(d+)(D+)(d+)(D+)(d+)(D+)(d+)(D+)(d+)(D?)`

D+ - 1..N znaků neobsahujících číslice

d+ - 1..N číslic

D? - 0..N znaků neobsahujících číslice

### 5.2.1.2 Obrázky

Aplikace umí rovněž načíst všechny běžně užívané formáty obrázků, pro které je napsána v .NET Frameworku podpora.

Obrázek musí mít rozměry dělitelné osmi, v opačném případě dojde ke změně velikosti obrázku na požadovanou velikost. Velikost obrázku může být maximálně 320x440. V případě většího rozlišení dojde ke změně velikosti obrázku za zachování pomětu stran a dodržení předchozí podmínky dělitelnosti osmi.

Načítání obrázků může být využito jako pomoc uživateli, kdy chce zkusit generovat otisk z nějakého skutečného otisku, který má uložený jako obrázek a nemá možnost nebo nechce rozpoznat markanty v jiné aplikaci. Na načteném obrázku otisku pak uživatel může ručně vkládat markanty na pozice dle vlastního uvážení.

## 5.2.2 Výstup aplikace

Aby se z aplikace daly získat nějaké výsledky a data a nešlo jen o vizuální výsledek za běhu aplikace, musí se data uložit v nějakém rozumném formátu.

Je samozřejmostí, že kdykoli během práce je možné uložit kreslicí plochu na disk jako obrázek, stejně tak se dají uložit i markanty, které právě aplikace používá jako data. Obrázky mohou sloužit pro porovnání jednotlivých nastavení a parametrů a sada markantů pro případné testování stejného vzorku markantů. Důležitou vlastností je, že ukládání obrázků není omezeno pouze na vygenerovaný obrazec papilárních linií, ale k uložení může dojít kdykoli se nám to hodí. Lze tak tedy jednoduše uložit nejen výsledek práce, ale například i rozložení hustoty, které právě používáme, pole orientací a vizuální rozložení markantů.

## 5.3 Nastavitelné parametry

Aby aplikace byla maximálně variabilní, lze nastavovat i hodnoty pro výpočet gaborova filtru, kdy uživatel bude moci testovat různé filtry a jejich vliv na výsledný vygenerovaný otisk.

Mezi nastavitelné věci uživatelem tedy patří:

- Gaborův filter
- Hustota rozložení
- Viditelná plocha otisku prstu
- Markanty

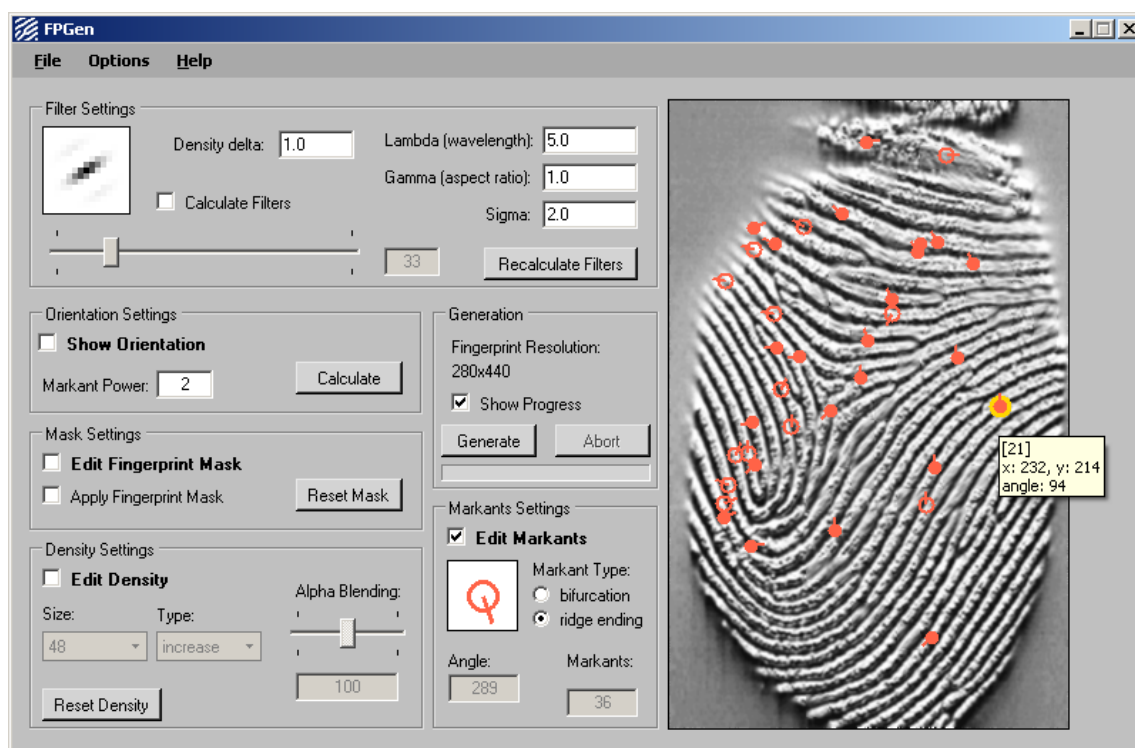
O podrobnější práci s aplikací a ovládáním se dočtete v příloze manuál, která je v digitální podobě uložena na CD..

## 5.4 Výsledná aplikace a zhodnocení výsledků

Aplikace byla vzhledově a chováním navrhována tak, aby byla co nejpřehlednější a aby se co nejjednodušeji ovládala.

Pro popisky jednotlivých komponent a funkcí byl zvolen anglický jazyk, aby byla aplikace univerzálnější. Většina výrazů je ovšem pochopitelná i pro laika a v problematice otisků prstů se nejedná o žádné špatně odhadnutelné pojmy, které by svým významem nedávaly smysl.

Vzhled aplikace je zobrazen na obrázku 5.1, kde jsou vidět kompletní možnosti nastavení a chování aplikace (obrázek otisku je načtený obrázek, markanty již zpracovány z textového souboru). Uživatel tedy nebude muset procházet několika okny, kde by nastavoval jednotlivé věci, ale může nastavit dle uvážení to, co zrovna potřebuje.



Obrázek 5.1: Vzhled aplikace

V pravé části okna aplikace, kde je zobrazen otisk, se provádí veškeré grafické operace a výsledné generování. Levá část slouží pro nastavování a výběr právě požadované činnosti, kterou chceme s aplikací dělat.

Vlastní generování se v aplikaci provádí v samostatném vlákne, aby se zamezilo vytíženosti celé aplikace a hlavně grafického uživatelského rozhraní a nemuselo se čekat až skončí generování. Tím nedojde k přerušení interakce mezi uživatelem a aplikací a zároveň je možnost zobrazit aktuální stav výpočtu celého otisku v procentech, takže má uživatel ještě představu o případné době trvání celého procesu.

Během generování jsou samozřejmě některé funkce aplikace omezeny, aby nedocházelo mezi vlákem výpočtu a vlákem samotné aplikace ke kolizím, kdy by uživatel změnil některá data, která se během výpočtu užívají.

Na obrázku 5.2 je pak vidět ukázka vygenerování malého obrázku. Vlevo bez vymezení zobrazené části a vpravo už s vymezením okrajů.



**Obrázek 5.2: Ukázka výsledku generování**

Díky podstatě metody užívání gaborova filtru dochází při generování otisku k vytváření dalších markantů na náhodné pozici, tomuto jevu však v této metodě nelze zabránit. Na výsledek to ale myslím nemá vliv, protože důležité je, aby vytvořený otisk obsahoval markanty, které zadal uživatel.

Aplikace výsledek generuje poměrně rychle. Při sledování průběhu vykreslování je však generování pomalejší díky převodům do zobrazitelné podoby obrázku.

## **5.5 Návrhy pro další možnosti aplikace a její další vývoj**

Jako většina složitějších aplikací má mnoho verzí a jejich vývoj pokračuje dál a dál za účelem zlepšování a zdokonalování samotné aplikace a algoritmu, tak i na tomto projektu se najde spousta věcí, které by se daly vylepšovat.

Výstupem této aplikace je binarizovaný obrázek reprezentující papilární linie. Tento obrázek by se aplikací dalších filtrů mohl upravit na podobu reálnějšího otisku prstu, což by nemělo patřit mezi náročné úkoly.

Možné úpravy binarizovaného otisku prstu:

- přidání rozostření hran a nepravidelnost papilárních linií
- zesvětlení otisku v blízkosti masky viditelnosti reprezentující 3D vzhled prstu a nestejnou sílu působící na jednotlivé části prstu při zanechání otisku
- přidání jizev
- zakomponování pozadí
- vylepšení algoritmu vytvářející pole orientací pro věrnější ztvárnění otisku

## 6 Závěr

Algoritmů a programů pro rozpoznávání otisků je mnoho, avšak pro generování otisků prstů jsem našel oficiálně ve světě pouze jeden. Tento program navíc nemá volně dostupné zdrojové kódy, algoritmus je však dobře zdokumentován a prezentován veřejnosti.

Po prozkoumání různých metod a úskalí, se kterými se setkáme při rozpoznávání otisků prstů jsme schopni porozumět i problematice pracující na opačném principu a to při generování samotného otisku. Svým způsobem máme práci zjednodušenou oproti rozpoznávání otisků, protože nemusíme žádný otisk porovnávat, ale vytváříme ho podle jistých pravidel. V konečném důsledku tedy učíme počítač, jak vytvořit syntetický otisk prstu.

Cílem projektu je implementace uživatelsky příjemného prostředí, pomocí kterého toto generování otisků provádí.

Zadáním tohoto projektu je generování otisků prstů pomocí předem zadaných markantů. Po prostudování dostupných materiálů jsem zjistil, že generování otisků s požadavkem na přesné souřadnice markantů ve vygenerovaném obrázku otisku prstu není vůbec snadným úkolem a předkládá mnoho problémů, které se kvůli tomu musí řešit.

Výsledkem práce je aplikace vytvořená v programovacím jazyce C# pracující pod platformou .NET Framework, která uživateli umožní snadné a přehledné nastavování možností generování otisků prstů, které se dají posléze uložit ve volitelném formátu. Jako u každé první verze rozsáhlejší aplikace je možný výskyt určitých chyb, ale ty během testování nebyly patrné. Aplikace umožňuje uživateli zadat pozice markantů co nejjednodušším a editovatelným způsobem a následně dokáže vygenerovat podle souřadnic těchto markantů odpovídající obrázek reprezentující průchody papilárních linií.

Podstatou algoritmu pro generování je obdobně jako u metody SFinGe sada gaborových filtrů, které jsou aplikovány na výstupní obrázek a postupně vzor papilárních linií vytváří. V aplikaci je kladen důraz na co největší variabilitu a možnosti, proto je možné nastavit parametry gaborova filtru, který bude použit pro samotné generování. Tím se dá v aplikaci provádět i jisté testování této metody generování otisků.

Možnost generování vlastních otisků prstů na základě pozic markantů v sobě skýtá určité varování. Již víme, že vygenerování takových otisků je možné a také víme, že existují rozpoznávací algoritmy, které pro porovnávání otisků prstů používají pouze souřadnice a parametry markantů. V tomto se pro přístroje používající tyto rozpoznávací metody objevuje jisté riziko, kdy by mohl být vygenerován falešný otisk na základě dat získaných odjinud a následně zneužit. Poslední dobou se ve



světě rozšiřují pasy s biometrickými údaji dané osoby. V České republice se jich v brzké chvíli také dočkáme. Pasy prozatím ponesou hlavně digitální podobu fotky majitele a jeho otisky prstů. Data na pasu jsou samozřejmě chráněna proti neoprávněnému čtení, avšak v dnešní době jsou již známy případy, kdy došlo k prolomení této ochrany. Údaje na datovém nosiči však naštěstí nejdou měnit a neoprávněně vytvářet, čímž by vyvstalo daleko více problémů.

Rozpoznávání otisků pouze podle markantů je algoritmicky nejjednodušší, avšak díky své podstatě nejvíce nebezpečné, což tento projekt dokázal možností generování vlastních otisků na základě těchto údajů.

Biometrické systémy však poslední dobou stále prochází vývojem a vytvářejí se stále sofistikovanější algoritmy a způsoby bezchybné identifikace osoby. Co se týče otisků prstů, tak se stále vyvíjí algoritmy pro větší zajištění bezpečnosti a spolehlivosti pro identifikaci osoby a rozpoznání neživého nebo synteticky vytvořeného prstu.

Podle kapitoly zabývající se metodou SFinGe v této práci se dá poměrně snadno implementovat generátor otisků prstů pracující na základě tvaru otisku a jeho singularit, což by mohlo být v budoucnu dalším rozšířením této aplikace. Generování takových otisků by sloužilo spíše pro testování rozpoznávacích algoritmů a sestavování databází otisků pro tyto algoritmy. Do aplikace by se dalo dále zahrnout i rozpoznávání markantů, čímž by se stala univerzálnější. Tato aplikace vytváří černobílý průběh papilárních linií. Pro tyto obrázky by tedy dále mohly být vytvořeny filtry, díky kterým by se zvýšila realističnost takto vygenerovaných otisků.

Téma této diplomové práce bylo značně specifické a proto nebylo zapotřebí studovat velké množství literatury. V českém jazyce je toto první práce, která se problematikou generování otisků prstů zabývá. Doufám tedy, že bude přínosem pro další lidi, kteří se této problematice budou chtít věnovat a tyto metody dále rozvíjet.

Všechny body zadání diplomové práce byly z pohledu autora splněny.

# Literatura

- [1] Straus, J., Porada, V. a kol. *Kriminalistická daktyloskopie*. Praha, PA ČR 2005.
- [2] Jain, L.C., Halici, U., Hayashi, I., Lee, S.B., Tsutsui, S. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. CRC Press, USA, 1999.
- [3] Cappeli, R., Erol, A., Maio, D., Maltoni, D. *Synthetic Fingerprint-image Generation*. ICPR2000, Barcelona, vol.3, pp.475-478, 2000.
- [4] Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S. *Handbook of Fingerprint Recognition*. Springer, New York, 2003
- [5] Maltoni, D. *A Tutorial on Fingerprint Recognition*, Biometric Systems Laboratory
- [6] WWW stránky. Fingerprint.  
<http://biolab.csr.unibo.it/Research.asp>. Vyhledáno v lednu 2007.
- [7] WWW stránky. Gaborovy filtry.  
[http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection\\_parameters.html](http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection_parameters.html). Vyhledáno v březnu 2007.
- [8] WWW stránky. Funkce.  
<http://local.wasp.uwa.edu.au/~pbourke/other/functions/>. Vyhledáno v dubnu 2007.
- [9] WWW stránky. C#.  
<http://cs.wikipedia.org/wiki/Csharp>. Vyhledáno v květnu 2007
- [10] Rajský, L., Seifert, A., Identifikace pomocí otisků prstů. <http://www.cs.vsb.cz/kratky/courses/2003-04/dis/reference/raj029sei041.pdf>, 2003
- [11] WWW stránky. Seriál Věda na stopě zločinu.  
<http://www.21století.cz/view.php?cisloclanku=2006012038>. Vyhledáno v květnu 2007

# Seznam příloh

Příloha 1. Manuál – nachází se v digitální podobě na DVD

Příloha 2. Zdrojové texty – kompletní projekt v jazyce C# se nachází na DVD

Příloha 3. DVD

# Seznam obrázků

OBRÁZEK 2.1: ZÁKLADNÍ TYPY MARKANTŮ .....	6
OBRÁZEK 2.2: ČÁST OTISKU S DETEKOVANÝMI MARKANTY .....	7
OBRÁZEK 2.3: ZÁKLADNÍ TYPY OTISKŮ PRSTŮ PODLE ROZLOŽENÍ SINGULARIT .....	9
OBRÁZEK 3.1: POLOHA A ÚHEL U MARKANTU TYPU ZAKONČENÍ A TYPU VIDLICE .....	11
OBRÁZEK 3.2: VYTVOŘENÍ POLE ORIENTACÍ Z PAPILÁRNÍCH LINÍ.....	12
OBRÁZEK 3.3: HUSTOTA ROZLOŽENÍ U DVOU RŮZNÝCH OTISKŮ PRSTŮ .....	12
OBRÁZEK 3.4: DETEKCE SINGULARIT POMOCÍ POINCARÉOVA INDEXU .....	14
OBRÁZEK 3.5: PŘÍKLAD DETEKCE SINGULARIT NA OTISKU PRSTU .....	14
OBRÁZEK 3.6: UŽITÍ GABOROVÝCH FILTRŮ PRO BINARIZACI.....	16
OBRÁZEK 3.7: ÚPRAVY OTISKU PRO DETEKCI MARKANTŮ .....	17
OBRÁZEK 3.8: DETEKCE MARKANTŮ POSTUPNÝM ZPRACOVÁNÍM OTISKU .....	17
OBRÁZEK 3.9: PŘÍMÁ DETEKCE MARKANTŮ Z OBRÁZKU VE STUPNÍCH ŠEDÍ.....	18
OBRÁZEK 3.10: POROVNÁNÍ OTISKŮ.....	19
OBRÁZEK 4.1: SFINGE – DEFINOVÁNÍ TVARU OTISKU .....	21
OBRÁZEK 4.2: SFINGE – UKÁZKA POLE ORIENTACÍ PODLE TŘÍDY OTISKU .....	22
OBRÁZEK 4.3: SFINGE – GABORŮV FILTR POUŽÍVANÝ V TÉTO METODĚ.....	24
OBRÁZEK 4.4: SFINGE – VYGENEROVANÉ OTISKY PRSTŮ .....	24
OBRÁZEK 4.5: VÝPOČET ORIENTACE .....	26
OBRÁZEK 4.6: ROZDÍL SÍLY VÁHOVÁNÍ.....	27
OBRÁZEK 4.7: REPREZENTACE ROZLOŽENÍ HUSTOTY .....	28
OBRÁZEK 4.8: REPREZENTACE ROZLOŽENÍ HUSTOTY POMOCÍ BAREV.....	29
OBRÁZEK 4.9: ZOBRAZENÍ KŘIVEK.....	30
OBRÁZEK 4.10: REPREZENTACE GABOROVA FILTRU.....	31
OBRÁZEK 4.11: GABORŮV FILTR - VLNOVÁ DÉLKA .....	32
OBRÁZEK 4.12: GABORŮV FILTR - ORIENTACE .....	32
OBRÁZEK 4.13: GABORŮV FILTR – PROSTOROVÝ POMĚR .....	32
OBRÁZEK 4.14: GENEROVÁNÍ OD POZICE MARKANTU.....	35
OBRÁZEK 4.15: GENEROVÁNÍ OD VÍCE MARKANTŮ NAJEDNOU .....	35
OBRÁZEK 5.1: VZHLED APLIKACE .....	40
OBRÁZEK 5.2: UKÁZKA VÝSLEDKU GENEROVÁNÍ.....	41